

The image is a large, symmetrical, abstract graphic composed of the letters 'S' and 'Y' arranged in a pattern that resembles a stylized 'H' or a pair of wings. The letters are black on a white background. The central vertical column is the tallest, flanked by shorter columns, and the outermost columns are the shortest. The overall shape is roughly rectangular with a wide base and a narrower top.

```
SSSSSSSS  WW      WW      AAAAAA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SSSSSSSS  WW      WW      AAAAAA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SS      WW      WW      AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SS      WW      WW      AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SS      WW      WW      AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SSSSSS  WW      WW      AA      AA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SSSSSS  WW      WW      AA      AA  PPPPPPPP  PPPPPPPP  EEEEEEEEE  RRRRRRRR
SS      WW  WW  WW  AAAAAAAAAA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SS      WW  WW  WW  AAAAAAAAAA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SS      WWW  WWW  AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SS      WWW  WWW  AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SSSSSSSS  WW      WW      AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
SSSSSSSS  WW      WW      AA      AA  PP      PP  PP      PP  EEEEEEEEE  RR      RR
                                     ....
                                     ....
                                     ....
                                     ....

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

(2)	231	DECLARATIONS
(9)	885	EX\$SWAPINIT - INITIALIZATION AND STARTUP FOR SWAPPER
(10)	1078	SWAPPER - MAIN LOOP
(11)	1105	BALANCE FREE PAGE COUNT
(12)	1152	SCHEDULE SWAP
(13)	1222	OUTSWAP
(16)	1479	RELPHD - RELEASE PROCESS HEADER
(17)	1573	DELPHD - DELETE PROCESS HEADER FOR DELETED PROCESS
(18)	1619	GBLTRANS/GBLVALID/GBLWRTVALID - HANDLE GLOBAL PAGES
(19)	1699	PROCTrans - PROCESS PAGE IN TRANSITION
(20)	1751	PAGE TABLE WORKING SET LIST ENTRIES
(21)	1767	IN\$SWAP
(24)	2257	FILLPHD - FILL SPT ENTRIES TO MAP PHD
(25)	2313	RELINIT - INITIALIZE REGISTERS FOR PAGE RELEASE LOOP
(26)	2342	OSINIT - OUTSWAP SCAN REGISTER INITIALIZATION
(27)	2366	RELPAGE - RELEASE DUPLICATE PAGE
(28)	2401	SWPREAD/SWPWRITE - SWAPPER I/O ROUTINES



```
0000 1 .TITLE SWAPPER WORKING SET SWAPPER
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *****
0000 26
0000 27 ++
0000 28 FACILITY: EXECUTIVE, SWAPPER
0000 29
0000 30 ABSTRACT: THE SWAPPER SCHEDULES AND EXECUTES SWAPPING OF PROCESS
0000 31 WORKING SETS BETWEEN SWAP STORAGE AND MAIN MEMORY.
0000 32
0000 33 ENVIRONMENT:
0000 34 MODE = KERNEL , RESIDENT
0000 35
0000 36 AUTHOR: R. HUSTVEDT CREATION DATE: 30-NOV-76
0000 37
0000 38 MODIFIED BY:
0000 39
0000 40 V03-029 ACG0440 Andrew C. Goldstein, 24-Jul-1984 10:50
0000 41 Add ref count field to ORB
0000 42
0000 43 V03-028 LMP0275 L. Mark Pilant, 12-Jul-1984 20:31
0000 44 Initialize the ACL info in the ORB to be a null descriptor
0000 45 list rather than an empty queue. This avoids the overhead
0000 46 of locking and unlocking the ACL mutex, only to find out
0000 47 that the ACL was empty.
0000 48
0000 49 V03-027 TMK0011 Todd M. Katz 11-Apr-1984
0000 50 The ACL mutexes within the Object Rights Blocks of the system
0000 51 and system directory logical name tables are currently
0000 52 incorrectly initialized to ^X00001111. Initialize them to
0000 53 ^X0000FFFF.
0000 54
0000 55 V03-026 MSH0029 Michael S. Harvey 9-Apr-1984
0000 56 The translation of LNMS$TEMPORARY_MAILBOX will now be LNMS$JOB
0000 57 instead of LNMS$GROUP. This is a part of an effort to close
```

0000 58 :  
0000 59 :  
0000 60 :  
0000 61 :  
0000 62 :  
0000 63 :  
0000 64 :  
0000 65 :  
0000 66 :  
0000 67 :  
0000 68 :  
0000 69 :  
0000 70 :  
0000 71 :  
0000 72 :  
0000 73 :  
0000 74 :  
0000 75 :  
0000 76 :  
0000 77 :  
0000 78 :  
0000 79 :  
0000 80 :  
0000 81 :  
0000 82 :  
0000 83 :  
0000 84 :  
0000 85 :  
0000 86 :  
0000 87 :  
0000 88 :  
0000 89 :  
0000 90 :  
0000 91 :  
0000 92 :  
0000 93 :  
0000 94 :  
0000 95 :  
0000 96 :  
0000 97 :  
0000 98 :  
0000 99 :  
0000 100 :  
0000 101 :  
0000 102 :  
0000 103 :  
0000 104 :  
0000 105 :  
0000 106 :  
0000 107 :  
0000 108 :  
0000 109 :  
0000 110 :  
0000 111 :  
0000 112 :  
0000 113 :  
0000 114 :

some privilege related security holes involving logical names and temporary mailbox creation.

V03-025 TMK0010 Todd M. Katz 26-Mar-1984  
Modify the logical name system services to make use of the updated internal protection checking mechanisms. What this involves is replacing the system directory and system logical name tables' CHIP protection templates with quad-word aligned Object Rights Blocks.

V03-024 TMK0009 Todd M. Katz 07-Mar-1984  
Add a hash code field, LNM\$W\_HASH, to every translation block of every logical name table template defined. This hash code field will be used in an optimization of logical name table name PROCESSING.

V03-023 LY00b7 Larry Yetto 16-FEB-1984 14:33  
Fix alignment of logical name tables

V03-022 ROW62094 Ralph O. Weber 25-JAN-1984  
Add PROCESSING for inswapped global page when there currently exists a equivalent global page having a page read error. This makes the list of possible conditions to be handled for an inswap of a global page: 1) no equivalent global page exists, 2) an equivalent global page exists, 3) the equivalent page is still being read (from a page fault read), 4) the equivalent page was read but encountered a page read error.

V03-021 TMK0008 Todd M. Katz 06-Jan-1984  
Never allow the system directory logical name table to be deleted. This is done as follows:  

1. Set the LNMB\$V\_NODELETE bit within the LNMB\$B\_FLAGS field of the system directory logical name table.
2. Check for this bit within the logical name system services whenever a LNMB is to be deleted.
3. If this bit is set, do not allow the LNMB to be deleted; otherwise, proceed with the deletion.

This mechanism will prevent the directories from ever being explicitly or implicitly deleted which can cause all sorts of problems.

V03-020 TMK0007 Todd M. Katz 25-Dec-1983  
Make a small change to TMK0006. Setup the remaining quota byte field of the system directory logical name table with a value of positive infinity (i.e. - ^X7FFFFFFF) instead of a value of positive infinity minus the size of the system table. This is necessary because the routine which is used to appropriately insert the system table performs the necessary quota subtractions; thus, in TMK0006 quota for the system table was being subtracted twice from the system directory logical name table.

V03-019 TMK0006 Todd M. Katz 18-Dec-1983  
Handcraft the system logical name table, LNM\$SYSTEM TABLE, instead of using the system service (\$CRELNT) to CREATE it.



0000 115 :  
0000 116 :  
0000 117 :  
0000 118 :  
0000 119 :  
0000 120 :  
0000 121 :  
0000 122 :  
0000 123 :  
0000 124 :  
0000 125 :  
0000 126 :  
0000 127 :  
0000 128 :  
0000 129 :  
0000 130 :  
0000 131 :  
0000 132 :  
0000 133 :  
0000 134 :  
0000 135 :  
0000 136 :  
0000 137 :  
0000 138 :  
0000 139 :  
0000 140 :  
0000 141 :  
0000 142 :  
0000 143 :  
0000 144 :  
0000 145 :  
0000 146 :  
0000 147 :  
0000 148 :  
0000 149 :  
0000 150 :  
0000 151 :  
0000 152 :  
0000 153 :  
0000 154 :  
0000 155 :  
0000 156 :  
0000 157 :  
0000 158 :  
0000 159 :  
0000 160 :  
0000 161 :  
0000 162 :  
0000 163 :  
0000 164 :  
0000 165 :  
0000 166 :  
0000 167 :  
0000 168 :  
0000 169 :  
0000 170 :  
0000 171 :

V03-018 WMC0018 Wayne Cardoza 02-Dec-1983  
PHD\$W\_WSLX, PHD\$W\_BAK have become longwords.

V03-017 TMK0005 Todd M. Katz 19-Oct-1983  
Add the following kernel mode logical names which will be  
used in order to optimize \$TRNLOGs:

TRNLOG\$\_PROCESS\_GROUP  
TRNLOG\$\_PROCESS\_SYSTEM  
TRNLOG\$\_GROUP\_SYSTEM  
TRNLOG\$\_PROCESS\_GROUP\_SYSTEM

Also, CREATE LNM\$TEMPORARY\_MAILBOX with a translation of  
LNM\$GROUP instead of LNM\$JOB.

V03-016 TMK0004 Todd M. Katz 11-Oct-1983  
Make the following changes to the logical names and tables  
that are CREATED at system initialization time:

1. CREATE LNM\$FILE\_DEV with the translations  
LNM\$PROCESS, LNM\$JOB, LNM\$GROUP, LNM\$SYSTEM (the change is the  
addition of the LNM\$JOB translation).
2. CREATE LNM\$TEMPORARY\_MAILBOX with the translation LNM\$JOB.
3. Remove LNM\$TRNLOG\_PG, LNM\$TRNLOG\_PS, LNM\$TRNLOG\_GS,  
LNM\$TRNLOG\_PGS, and LNM\$DEFAULT\_SEARCH.
4. Add the following kernel mode logical names which will be  
used in order to provide compatibility between V3 and V4  
for all of the old logical name system services (\$TRNLOG,  
\$CRELOG, \$DELLOG):

LOG\$PROCESS  
LOG\$GROUP  
LOG\$SYSTEM

5. Add to the PQB\$AB\_SYSPQL quota list a PQL\$\_JTQUOTA item.

V03-015 TMK0003 Todd M. Katz 09-Apr-1983  
Statically define the CHIP protection structure of  
LNM\$SYSTEM DIRECTORY, set the (internal) attribute bit  
LNM\$V\_SYSTEM when creating LNM\$SYSTEM TABLE, and change the  
CHIP protection of LNM\$SYSTEM DIRECTORY and LNM\$SYSTEM TABLE to  
S:RWE O:RWE G:R W:R. Also, CREATE the supervisor mode logical  
name LNM\$FILE\_DEV with the translations LNM\$PROCESS, LNM\$GROUP,  
LNM\$SYSTEM instead of LNM\$DEFAULT\_SEARCH, mark both  
translations of LNM\$DIRECTORIES with the TERMINAL attribute,  
and CREATE the non-aliasable kernel mode logical names  
LNM\$TRNLOG\_PG, LNM\$TRNLOG\_PS, LNM\$TRNLOG\_GS, LNM\$TRNLOG\_PGS.

V03-014 KDM0052 Kathleen D. Morse 11-Jul-1983  
Replace references of PR\$\_TODR with EXE\$GQ\_SYSTIME+2.

V03-013 DMW4060 DMWalp 23-Jun-1983  
Change \$xxLNM value parameters to be by reference

V03-012 DMW4054 DMWalp 21-Jun-1983  
Convert SYS\$DISK and SYS\$SYSDEVICE creation from \$CRELOG

0000 172 :  
0000 173 :  
0000 174 :  
0000 175 :  
0000 176 :  
0000 177 :  
0000 178 :  
0000 179 :  
0000 180 :  
0000 181 :  
0000 182 :  
0000 183 :  
0000 184 :  
0000 185 :  
0000 186 :  
0000 187 :  
0000 188 :  
0000 189 :  
0000 190 :  
0000 191 :  
0000 192 :  
0000 193 :  
0000 194 :  
0000 195 :  
0000 196 :  
0000 197 :  
0000 198 :  
0000 199 :  
0000 200 :  
0000 201 :  
0000 202 :  
0000 203 :  
0000 204 :  
0000 205 :  
0000 206 :  
0000 207 :  
0000 208 :  
0000 209 :  
0000 210 :  
0000 211 :  
0000 212 :  
0000 213 :  
0000 214 :  
0000 215 :  
0000 216 :  
0000 217 :  
0000 218 :  
0000 219 :  
0000 220 :  
0000 221 :  
0000 222 :  
0000 223 :  
0000 224 :  
0000 225 :  
0000 226 :  
0000 227 :  
0000 228 :--

to \$CRELNM

V03-011 RAS0158 Ron Schaefer 23-May-1983  
Add CHIP protection structure to the logical name structures.  
Protection stuff only supports SOGW checking for now.  
Fix quota for LNM\$SYSTEM\_TABLE.

V03-010 TMK0002 Todd M. Katz 26-Apr-1983  
CREATE the following logical name structures at system  
initialization time:

1. LNM\$SYSTEM\_TABLE.
2. LNM\$SYSTEM.
3. LNM\$FILE\_DEV (Executive Mode).
4. LNM\$FILE\_DEV (Supervisor Mode).
5. LNM\$DEFAULT\_SEARCH.
6. LNM\$TEMPORARY\_MAILBOX.
7. LNM\$PERMANENT\_MAILBOX.
8. LNM\$DIRECTORIES.

Change the name of LNT\$SYSTEM\_DIRECTORY to LNM\$SYSTEM\_DIRECTORY.

V03-009 TMK0001 Todd M. Katz 14-Apr-1983  
Make the following changes to the system directory logical  
name table:

1. Make the table a kernel access mode table.
2. Make LNMBSL\_TABLE point to the system directory table's  
table header.
3. Set the bits LNMTH\$V\_SHAREABLE and LNMTH\$V\_DIRECTORY within  
LNMTH\$B\_FLAGS.
4. Delete the field LNMTH\$SL\_LOGNAM.

V03-008 HRJ0200 Herb Jacobs 05-Feb-1983  
Add check to BALANCE to remove confusion as to why  
swapper has woken up. If there are FREELIM pages on  
freelist, then don't acquire FREEGOAL pages, but rather  
perform requested function woken up for.

V03-007 DMW4020 DMWalp 30-Dec-1982  
Added creation system logical directory.

V03-006 DMW4019 DMWalp 15-Dec-1982  
Calculate LNM hash table parameters and CREATE hash table.

V03-005 DMW4006 DMWalp 10-NOV-1982  
Recode creation SYS\$DISK and SYS\$SYSDEVICE to use  
external interface ( not internal ) of \$CRELOG

V03-004 HRJ0101 Herb Jacobs 30-Jun-1982  
Add perturbation to balance set slot scanner to try to  
alleviate deadlocks caused there if seemingly the  
best swapper action is to try to free PROCESS waited  
for service from an outswapped PROCESS.



```
0000 231      .SBTTL  DECLARATIONS
0000 232      :
0000 233      : INCLUDE FILES:
0000 234      :
0000 235
0000 236      $ACBDEF      ; DEFINE  AST CONTROL BLOCK OFFSETS
0000 237      $DYNDEF     ; DEFINE  STRUCTURE TYPE CODES
0000 238      $IPLDEF     ; DEFINE  INTERRUPT PRIORITY LEVELS
0000 239      $IRPDEF     ; DEFINE  I/O REQUEST PACKET OFFSETS
0000 240      $LNMDEF     ; DEFINE  LOGICAL NAME OFFSETS
0000 241      $LNMSTRDEF  ; DEFINE  LOGICAL NAME STRUCTURE OFFSETS
0000 242      $OPDEF      ; DEFINE  OPCODE EQUIVALENT VALUES
0000 243      $ORBDEF     ; DEFINE  OBJECT RIGHTS BLOCK OFFSETS
0000 244      $PCBDEF     ; DEFINE  PCB OFFSETS
0000 245      $PFLDEF     ; DEFINE  SWAP FILE TABLE OFFSETS
0000 246      $PFNDEF     ; DEFINE  PFN VALUES
0000 247      $PHDDEF     ; DEFINE  PHD OFFSETS
0000 248      $PQLDEF     ; DEFINE  QUOTA SYMBOLS
0000 249      $PRDEF      ; DEFINE  PROCESSOR REGISTERS
0000 250      $PRCDEF     ; CREATE  PROCESS FLAGS
0000 251      $PSLDEF     ; DEFINE  PSL VALUES
0000 252      $PTEDEF     ; DEFINE  PAGE TABLE ENTRY
0000 253      $VADEF      ; DEFINE  VIRTUAL ADDRESS FIELDS
0000 254      $WSLDEF     ; DEFINE  WORKING SET LIST BITS
0000 255
0000 256      :
0000 257      : ASSUMPTIONS ABOUT THE STRUCTURE OF LOGICAL NAME AND OBJECT RIGHTS BLOCKS:
0000 258      :
0000 259
0000 260      ASSUME  LNMB$$_FLINK,      EQ,  0
0000 261      ASSUME  LNMB$$_FLINK+4,   EQ,  LNMB$$_BLINK
0000 262      ASSUME  LNMB$$_BLINK+4,   EQ,  LNMB$$_SIZE
0000 263      ASSUME  LNMB$$_SIZE+2,    EQ,  LNMB$$_TYPE
0000 264      ASSUME  LNMB$$_TYPE+1,    EQ,  LNMB$$_ACMODE
0000 265      ASSUME  LNMB$$_ACMODE+1,  EQ,  LNMB$$_TABLE
0000 266      ASSUME  LNMB$$_TABLE+4,   EQ,  LNMB$$_FLAGS
0000 267      ASSUME  LNMB$$_FLAGS+1,   EQ,  LNMB$$_NAME
0000 268
0000 269      ASSUME  LNMX$$_FLAGS,      EQ,  0
0000 270      ASSUME  LNMX$$_FLAGS+1,   EQ,  LNMX$$_INDEX
0000 271      ASSUME  LNMX$$_INDEX+1,   EQ,  LNMX$$_HASH
0000 272      ASSUME  LNMX$$_HASH+2,    EQ,  LNMX$$_XLATION
0000 273
0000 274      ASSUME  LNMTH$$_FLAGS,     EQ,  0
0000 275      ASSUME  LNMTH$$_FLAGS+1,  EQ,  LNMTH$$_HASH
0000 276      ASSUME  LNMTH$$_HASH+4,    EQ,  LNMTH$$_ORB
0000 277      ASSUME  LNMTH$$_ORB+4,     EQ,  LNMTH$$_NAME
0000 278      ASSUME  LNMTH$$_NAME+4,    EQ,  LNMTH$$_PARENT
0000 279      ASSUME  LNMTH$$_PARENT+4,  EQ,  LNMTH$$_CHILD
0000 280      ASSUME  LNMTH$$_CHILD+4,   EQ,  LNMTH$$_SIBLING
0000 281      ASSUME  LNMTH$$_SIBLING+4, EQ,  LNMTH$$_QTABLE
0000 282      ASSUME  LNMTH$$_QTABLE+4,  EQ,  LNMTH$$_BYTESLM
0000 283      ASSUME  LNMTH$$_BYTESLM+4, EQ,  LNMTH$$_BYTES
0000 284
0000 285      ASSUME  ORB$$_OWNER,        EQ,  0
0000 286      ASSUME  ORB$$_OWNER+4,     EQ,  ORB$$_ACL_MUTEX
0000 287      ASSUME  ORB$$_ACL_MUTEX+4,  EQ,  ORB$$_SIZE
```



0000	288	ASSUME	ORB\$W_SIZE+2,	EQ,	ORB\$B_TYPE
0000	289	ASSUME	ORB\$B_TYPE+1,	EQ,	ORB\$B_FLAGS
0000	290	ASSUME	ORB\$B_FLAGS+3,	EQ,	ORB\$W_REFCOUNT
0000	291	ASSUME	ORB\$W_REFCOUNT+2,	EQ,	ORB\$Q_MODE_PROT
0000	292	ASSUME	ORB\$Q_MODE_PROT+8,	EQ,	ORB\$L_SYS_PROT
0000	293	ASSUME	ORB\$L_SYS_PROT+4,	EQ,	ORB\$L_OWN_PROT
0000	294	ASSUME	ORB\$L_OWN_PROT+4,	EQ,	ORB\$L_GRP_PROT
0000	295	ASSUME	ORB\$L_GRP_PROT+4,	EQ,	ORB\$L_WOR_PROT
0000	296	ASSUME	ORB\$L_WOR_PROT+4,	EQ,	ORB\$L_ACL_COUNT
0000	297	ASSUME	ORB\$L_ACL_COUNT+4,	EQ,	ORB\$L_ACL_DESC
0000	298	ASSUME	ORB\$L_ACL_DESC+4,	EQ,	ORB\$R_MIN_CLASS
0000	299	ASSUME	ORB\$R_MIN_CLASS+ORB\$S_MIN_CLASS,-		
0000	300			EQ,	ORB\$R_MAX_CLASS
0000	301	ASSUME	ORB\$R_MAX_CLASS+ORB\$S_MAX_CLASS,-		
0000	302			EQ,	ORB\$K_LENGTH

```
0000 304
0000 305 :
0000 306 : OWN STORAGE:
0000 307 :
0000 308 :
00000000 309 .PSECT $$$220, LONG : SWAPPER/SCHEDULER WRITABLE DATA
0000 310 IOROUTINE: : ADDRESS OF PROPER BUILD PACKET ROUTINE
00000000 0000 311 .LONG 0 :
00000000 0004 312 IOEA: .LONG 0 : I/O END ACTION RETURN
00000000 0008 313 RWSSWP: .LONG 0 : REMAINING WS SWP ADDRESS
00000000 000C 314 RSVAPTE: .LONG 0 : REMAINING SVA OF PTE
0000 0010 315 RPGCNT: .WORD 0 : REMAINING PAGE COUNT
0000 0012 316 OSWPPGS: .WORD 0 : OUTSWAP PAGE COUNT
00000000 0014 317 OSWPPCB: .LONG 0 : PCB ADDRESS OF OUTSWAP PROCESS
0018 318 SWPSGW_BALCNT:: : COUNT OF PROCESSES IN BALANCE SET
FFFF 0018 319 .WORD -1 : EXCLUDING NULL PROCESS AND SWAPPER
001A 320 SCHSGW_SWPFCNT:: : COUNT OF SUCCESSIVE SWAP
0000 001A 321 .WORD 0 : SCHEDULE FAILURES.
001C 322 :
00000000 323 .PSECT $$$260,5 : WRITABLE, HIGH USE PSECT
0000 324 :
0000 325 :
0000 326 : LNM$SYSTEM_DIRECTORY - THE SYSTEM DIRECTORY LOGICAL NAME TABLE.
0000 327 :
0000 328 :
0000 329 LNM$SYSTEM_DIRECTORY::
00000000 0000 330 .LONG 0 : FORWARD LINK
00000000 0004 331 .LONG 0 : BACK LINK
0000 0008 332 .WORD LNM SYS DIR_SIZ : SIZE OF STRUCTURE
40 000A 333 .BYTE DYN$C_LNM : TYPE OF STRUCTURE
00 000B 334 .BYTE PSL$C_KERNEL : KERNEL ACCESS MODE
0000002B 000C 335 .ADDRESS LNM_SYSTEM_DIR_LNMTH : DIRECTORY TABLE HEADER ADDRESS
19 0010 336 .BYTE LNMBSM_NO_ACIAST- : DIRECTORY TABLES CAN NOT BE ALIASED
0011 337 LNMBSM_TABLE!- : DIRECTORIES ARE TABLES
0011 338 LNMBSM_NODELETE : DIRECTORIES CAN NOT BE DELETED
SF 4D 45 54 53 59 53 24 4D 4E 4C 00 0011 339 .ASCII "LNM$SYSTEM_DIRECTORY" : NAME OF DIRECTORY TABLE
59 52 4F 54 43 45 52 49 44 001D
14 0011
0026 340
02 0026 341 .BYTE LNMX$M_TERMINAL : FLAGS BYTE. NO MORE TRANSLATIONS
82 0027 342 .BYTE LNMX$C_TABLE : TRANSLATION INDEX ( SPECIAL TABLE )
0000 0028 343 .WORD 0 : TRANSLATION HASH CODE
25 002A 344 .BYTE LNMTH$K_LENGTH : SIZE OF TABLE HEADER BLOCK
002B 345
002B 346 LNM_SYSTEM_DIR_LNMTH::
03 002B 347 .BYTE LNMTH$M_SHAREABLE!- : DIRECTORY IS A SHAREABLE TABLE
002C 348 LNMTH$M_DIRECTORY : TABLE IS A DIRECTORY TABLE
00000000 002C 349 .LONG 0 : ADDRESS OF HASH TABLE
00000058 0030 350 .ADDRESS LNM_SYSTEM_DIR_ORB : ADDRESS OF OBJECT RIGHTS BLOCK
00000000 0034 351 .ADDRESS LNM$SYSTEM_DIRECTORY : ADDRESS OF CONTAINING LNMB BLOCK
00000000 0038 352 .LONG 0 : ADDRESS OF PARENT TABLE
00000000 003C 353 .LONG 0 : ADDRESS OF CHILD TABLE
00000000 0040 354 .LONG 0 : ADDRESS OF SIBLING TABLE
0000002B 0044 355 .ADDRESS LNM_SYSTEM_DIR_LNMTH : ADDRESS OF TABLE HOLDING QUOTA
7FFFFFFF 0048 356 .LONG *X7FFFFFFF : INITIAL QUOTA ( POSITIVE INFINITY )
7FFFFFFF 004C 357 .LONG *X7FFFFFFF : REMAINING QUOTA ( POSITIVE INFINITY )
0050 358
```



	04	0050	359	.BYTE	LNM\$SM_XEND	: FLAGS BYTE. NO MORE TRANSLATIONS
		0051	360			
		0051	361	.ALIGN	QUAD	
		0058	362	LNM_SYSTEM_DIR_ORB:		
00010004	0058	363		.LONG	*X00010004	: SYSTEM DIRECTORY OWNER IS [1,4]
0000 FFFF	005C	364		.WORD	-1,0	: INITIALIZE ACL MUTEX
0068	0060	365		.WORD	LNM_SYS_DIR_ORB_SIZ	: SIZE OF OBJECT RIGHTS BLOCK
49	0062	366		.BYTE	DYN\$C_ORB	: BLOCK TYPE
00	0063	367		.BYTE	0	: NO ACL AS YET
00000000	0064	368		.LONG	0	: ZERO RESERVED WORD & REF COUNT
00000000	0068	369		.QUAD	0	: OBJECT DOES NOT HAVE AN ACCESS MODE
00000008	0070	370		.LONG	*X00000008	: SYSTEM PROTECTION IS RWE
00000008	0074	371		.LONG	*X00000008	: OWNER PROTECTION IS RWE
0000000E	0078	372		.LONG	*X0000000E	: GROUP PROTECTION IS R
0000000E	007C	373		.LONG	*X0000000E	: WORLD PROTECTION IS R
00000000	0080	374		.LONG	0,0	: NULL INITIAL ACL
00'00'00'00'00'00'00'00'00'00'00'00'00'	0088	375		.BYTE	0[ORB\$S_MIN_CLASS]	: MINIMUM CLASSIFICATION MASK
00'00'00'00'00'00'00'00'00'00'00'00'00'	0094					
00'00'00'00'00'00'00'00'00'00'00'00'00'	009C	376		.BYTE	0[ORB\$S_MAX_CLASS]	: MAXIMUM CLASSIFICATION MASK
00'00'00'00'00'00'00'00'00'00'00'00'00'	00A8					
	00B0	377				
	00B0	378		.ALIGN	5	
00000068	00C0	379		LNM_SYS_DIR_ORB_SIZ = . - LNM_SYSTEM_DIR_ORB		
000000C0	00C0	380		LNM_SYS_DIR_SIZ = . - LNM\$SYSTEM_DIRECTORY		
	00C0	381				
	00C0	382		:		
	00C0	383		:: LNM\$SYSTEM_TABLE - THE SYSTEM LOGICAL NAME TABLE.		
	00C0	384		:		
	00C0	385				
	00C0	386		SYSTEM_TABLE:		
00000000	00C0	387		.LONG	0	: FORWARD LINK
00000000	00C4	388		.LONG	0	: BACK LINK
00C0	00C8	389		.WORD	SYSTEM_TABLE_SIZE	: SIZE OF STRUCTURE
40	00CA	390		.BYTE	DYN\$C_LNM	: TYPE OF STRUCTURE
00	00CB	391		.BYTE	PSL\$C_KERNEL	: KERNEL ACCESS MODE
0000002B	00CC	392		.ADDRESS	LNM_SYSTEM_DIR_LNMTH	: DIRECTORY TABLE HEADER ADDRESS
09	00D0	393		.BYTE	LNM\$SM_NO_ALIAS!	: TABLE CAN NOT BE ALIASED
	00D1	394			LNM\$SM_TABLE	: TABLE
SF 4D 45 54 53 59 53 24 4D 4E 4C 00'	00D1	395		.ASCII	"LNM\$SYSTEM_TABLE"	: TABLE NAME
45 4C 42 41 54	00DD					
10	00D1					
	00E2	396				
02	00E2	397		.BYTE	LNM\$SM_TERMINAL	: FLAGS BYTE. NO MORE TRANSLATIONS
82	00E3	398		.BYTE	LNM\$SC_TABLE	: TRANSLATION INDEX ( SPECIAL TABLE )
0000	00E4	399		.WORD	0	: TRANSLATION HASH CODE
25	00E6	400		.BYTE	LNMTH\$K_LENGTH	: SIZE OF TABLE HEADER BLOCK
	00E7	401				
	00E7	402		SYSTEM_TABLE_LNMTH:		
09	00E7	403		.BYTE	LNMTH\$M_SHAREABLE!-	: TABLE IS SHAREABLE
	00E8	404			LNMTH\$M_SYSTEM	: THIS IS THE SYSTEM LOGICAL NAME TABLE
00000000	00E8	405		.LONG	0	: ADDRESS OF HASH TABLE
00000110	00EC	406		.ADDRESS	SYSTEM_TABLE_ORB	: ADDRESS OF OBJECT RIGHTS BLOCK
000000C0	00F0	407		.ADDRESS	SYSTEM_TABLE	: ADDRESS OF CONTAINING LNMB BLOCK
0000002B	00F4	408		.ADDRESS	LNM_SYSTEM_DIR_LNMTH	: ADDRESS OF PARENT TABLE
00000000	00F8	409		.LONG	0	: ADDRESS OF CHILD TABLE
00000000	00FC	410		.LONG	0	: ADDRESS OF SIBLING TABLE
0000002B	0100	411		.ADDRESS	LNM_SYSTEM_DIR_LNMTH	: ADDRESS OF TABLE HOLDING QUOTA

# WORKING SET SWAPPER DECLARATIONS

```
16-SEP-1984 01:19:37 VAX/VMS Macro V04-00
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1
```

Page 9  
(4)

SWAI  
V04[illegible]



```
0180 438
00000000 439 .PSECT YF$LOWUSE ; PAGED PSECT AT END OF SYS.EXE
0000 440
49 4E 49 53 59 53 00000008'010E0000' 0000 441 IMGDESC:.ASCID /SYSINIT.EXE/ ; SYSTEM INITIALIZATION PROCESS
      45 58 45 2E 54 000E
3A 30 41 50 4F 0000001B'010E0000' 0013 442 TTDESC:.ASCID /OPA0:/
      0020 443
      0020 444
      0020 445 : DESCRIPTORS AND CHARACTER STRING BUFFERS FOR THE LOGICAL NAME TABLE NAMES,
      0020 446 : LOGICAL NAMES, AND LOGICAL NAME EQUIVALENCE STRINGS THAT ARE CREATED AT
      0020 447 : SYSTEM INITIALIZATION TIME.
      0020 448 :
      0020 449
49 44 24 4D 4E 4C 00000028'010E0000' 0020 450 LNM_DIRECTORIES_DESC:
      53 45 49 52 4F 54 43 45 52 0020 451 .ASCID /LNMS$DIRECTORIES/
      0037 452
      0037 453 LNM_FILE_DEV_DESC:
49 46 24 4D 4E 4C 0000003F'010E0000' 0037 454 .ASCID /LNMS$FILE_DEV/
      56 45 44 5F 45 4C 0045
      004B 455
      004B 456 LNM_PERMANENT_MAILBOX_DESC:
45 50 24 4D 4E 4C 00000053'010E0000' 004B 457 .ASCID /LNMS$PERMANENT_MAILBOX/
4C 49 41 4D 5F 54 4E 45 4E 41 4D 52 0059
      58 4F 42 0065
      0068 458
      0068 459 LNM_SYSTEM_DESC:
      0000000A' 0068 460 .LONG LNM_SYSTEM_LENGTH
      0000017D' 006C 461 .ADDRESS LNM_SYSTEM
      0070 462
      0070 463 LNM_SYSTEM_DIRECTORY_DESC:
      00000014' 0070 464 .LONG LNM_SYSTEM_DIRECTORY_LENGTH
      00000187' 0074 465 .ADDRESS LNM_SYSTEM_DIRECTORY
      0078 466
      0078 467 LNM_TEMPORARY_MAILBOX_DESC:
45 54 24 4D 4E 4C 00000080'010E0000' 0078 468 .ASCID /LNMS$TEMPORARY_MAILBOX/
4C 49 41 4D 5F 59 52 41 52 4F 50 4D 0086
      58 4F 42 0092
      0095 469
      0095 470 LOG_G_DESC:
      00000009' 0095 471 .LONG LOG_GROUP_LENGTH
      000001AB' 0099 472 .ADDRESS LOG_GROUP
      009D 473
      009D 474 LOG_P_DESC:
      0000000B' 009D 475 .LONG LOG_PROCESS_LENGTH
      000001B4' 00A1 476 .ADDRESS LOG_PROCESS
      00A5 477
      00A5 478 LOG_S_DESC:
      0000000A' 00A5 479 .LONG LOG_SYSTEM_LENGTH
      000001BF' 00A9 480 .ADDRESS LOG_SYSTEM
      00AD 481
49 44 24 53 59 53 000000B5'010E0000' 00AD 482 SYS_DISK_DESC:
      4B 53 008B 483 .ASCID /SYS$DISK/
      00BD 484
      00BD 485 SYS_SYSDEVICE_DESC:
59 53 24 53 59 53 000000C5'010E0000' 00BD 486 .ASCID /SYS$SYSDEVICE/
```

```

      45 43 49 56 45 44 53 00CB
      00D2 487
      00D2 488 TRNLOG_GS_DESC:
      00D2 489 .ASCID /TRNLOG$_GROUP_SYSTEM/
      00E0
      00EC
      00EE 490
      00EE 491 TRNLOG_PG_DESC:
      00EE 492 .ASCID /TRNLOG$_PROCESS_GROUP/
      00FC
      0108
      010B 493
      010B 494 TRNLOG_PS_DESC:
      010B 495 .ASCID /TRNLOG$_PROCESS_SYSTEM/
      0119
      0125
      0129 496
      0129 497 TRNLOG_PGS_DESC:
      0129 498 .ASCID /TRNLOG$_PROCESS_GROUP_SYSTEM/
      0137
      0143
      014D 499
      014D 500 LNM_GROUP:
      014D 501 .ASCII /LNM$GROUP/
      0156 502 LNM_GROUP_LENGTH = . - LNM_GROUP
      0156 503
      0156 504 LNM_JOB:
      0156 505 .ASCII /LNM$JOB/
      015D 506 LNM_JOB_LENGTH = . - LNM_JOB
      015D 507
      015D 508 LNM_PROCESS:
      015D 509 .ASCII /LNM$PROCESS/
      0168 510 LNM_PROCESS_LENGTH = . - LNM_PROCESS
      0168 511
      0168 512 LNM_PROCESS_DIRECTORY:
      0168 513 .ASCII /LNM$PROCESS_DIRECTORY/
      0174
      017D 514 LNM_PROCESS_DIRECTORY_LENGTH = . - LNM_PROCESS_DIRECTORY
      017D 515
      017D 516 LNM_SYSTEM:
      017D 517 .ASCII /LNM$SYSTEM/
      0187 518 LNM_SYSTEM_LENGTH = . - LNM_SYSTEM
      0187 519
      0187 520 LNM_SYSTEM_DIRECTORY:
      0187 521 .ASCII /LNM$SYSTEM_DIRECTORY/
      0193
      019B 522 LNM_SYSTEM_DIRECTORY_LENGTH = . - LNM_SYSTEM_DIRECTORY
      019B 523
      019B 524 LNM_SYSTEM_TABLE:
      019B 525 .ASCII /LNM$SYSTEM_TABLE/
      01A7
      01AB 526 LNM_SYSTEM_TABLE_LENGTH = . - LNM_SYSTEM_TABLE
      01AB 527
      01AB 528 LOG_GROUP:
      01AB 529 .ASCII /LOG$GROUP/
      01B4 530 LOG_GROUP_LENGTH = . - LOG_GROUP
      01B4 531
```



```
53 53 45 43 4F 52 50 24 47 4F 4C 01B4 532 LOG_PROCESS:
0000000B 01B4 533 .ASCII /LOG$PROCESS/
01BF 534 LOG_PROCESS_LENGTH = . - LOG_PROCESS
01BF 535
01BF 536 LOG_SYSTEM:
4D 45 54 53 59 53 24 47 4F 4C 01BF 537 .ASCII /LOG$SYSTEM/
0000000A 01C9 538 LOG_SYSTEM_LENGTH = . - LOG_SYSTEM
01C9 539
01C9 540 :
01C9 541 : ATTRIBUTE, ACCESS MODE AND ITEM BUFFERS WHICH ARE PASSED BY REFERENCE.
01C9 542 :
01C9 543
00000001 01C9 544 EXEC_MODE: : EXECUTIVE ACCESS MODE BUFFER
01C9 545 .LONG PSL$C_EXEC
01CD 546
00000000 01CD 547 KERNEL_MODE: : KERNEL ACCESS MODE BUFFER
01CD 548 .LONG PSL$C_KERNEL
01D1 549
00000002 01D1 550 SUPER_MODE: : SUPERVISOR ACCESS MODE BUFFER
01D1 551 .LONG PSL$C_SUPER
01D5 552
00000001 01D5 553 LNM_NO_ALIAS: : NO_ALIAS ATTRIBUTE BUFFER
01D5 554 .LONG LNM$M_NO_ALIAS
01D9 555
00000200 01D9 556 TERMINAL_BUFFER: : TERMINAL ATTRIBUTES ITEM BUFFER
01D9 557 .LONG LNM$M_TERMINAL
01DD 558
01DD 559 :
01DD 560 : ITEM LISTS FOR THE CREATION OF THE LOGICAL NAMES SETUP AT SYSTEM
01DD 561 : INITIALIZATION TIME.
01DD 562 :
01DD 563
0004 01DD 564 DIRECTORIES LIST: : ITEM LIST FOR LNM$DIRECTORIES
0003 01DD 565 .WORD 4 : TERMINAL ATTRIBUTES ITEM
000001D9 01DF 566 .WORD LNM$ ATTRIBUTES
00000000 01E1 567 .ADDRESS TERMINAL_BUFFER
01E5 568 .LONG 0
01E9 569
0015 01E9 570 .WORD LNM_PROCESS_DIRECTORY_LENGTH : LNM$PROCESS_DIRECTORY STRING ITEM
0002 01EB 571 .WORD LNM$ STRING
00000168 01ED 572 .ADDRESS LNM_PROCESS_DIRECTORY
00000000 01F1 573 .LONG 0
01F5 574
0014 01F5 575 .WORD LNM_SYSTEM_DIRECTORY_LENGTH : LNM$SYSTEM_DIRECTORY STRING ITEM
0002 01F7 576 .WORD LNM$ STRING
00000187 01F9 577 .ADDRESS LNM_SYSTEM_DIRECTORY
00000000 01FD 578 .LONG 0
0201 579
00000000 0201 580 .LONG 0 : END OF ITEM LIST
0205 581
000B 0205 582 FILE_DEV_SUPER_LIST: : ITEM LIST FOR SUPERVISOR LNM$FILE_DEV
0002 0207 583 .WORD LNM_PROCESS_LENGTH : LNM$PROCESS STRING ITEM
0000015D 0209 584 .WORD LNM$ STRING
00000000 020D 585 .ADDRESS LNM_PROCESS
0211 586 .LONG 0
0007 0211 587
0211 588 .WORD LNM_JOB_LENGTH : LNM$JOB STRING ITEM
```

```
0002 0213 589 .WORD LNM$ STRING
00000156' 0215 590 .ADDRESS LNM_JOB
00000000 0219 591 .LONG 0
0000 021D 592
0009 021D 593 .WORD LNM GROUP LENGTH ; LNM$GROUP STRING ITEM
0002 021F 594 .WORD LNM$ STRING
0000014D' 0221 595 .ADDRESS LNM_GROUP
00000000 0225 596 .LONG 0
0000 0229 597
000A 0229 598 .WORD LNM SYSTEM LENGTH ; LNM$SYSTEM STRING ITEM
0002 022B 599 .WORD LNM$ STRING
0000017D' 022D 600 .ADDRESS LNM_SYSTEM
00000000 0231 601 .LONG 0
00000000 0235 602
00000000 0235 603 .LONG 0 ; END OF ITEM LIST
0000 0239 604
0000 0239 605 FILE DEV EXEC LIST: ; ITEM LIST FOR EXECUTIVE LNM$FILE DEV
0000 0239 606 PERMANENT MAILBOX LIST: ; ITEM LIST FOR LNM$PERMANENT_MAILBOX
000A 0239 607 .WORD LNM SYSTEM LENGTH ; LNM$SYSTEM STRING ITEM
0002 023B 608 .WORD LNM$ STRING
0000017D' 023D 609 .ADDRESS LNM_SYSTEM
00000000 0241 610 .LONG 0
00000000 0245 611
00000000 0245 612 .LONG 0 ; END OF ITEM LIST
0000 0249 613
0009 0249 614 LOG_G_LIST: ; ITEM LIST FOR LOG$GROUP
0002 024B 615 .WORD LNM GROUP LENGTH ; LNM$GROUP STRING ITEM
0000014D' 024D 616 .WORD LNM$ STRING
00000000 024D 617 .ADDRESS LNM_GROUP
00000000 0251 618 .LONG 0
00000000 0255 619
00000000 0255 620 .LONG 0 ; END OF ITEM LIST
0000 0259 621
000B 0259 622 LOG_P_LIST: ; ITEM LIST FOR LOG$PROCESS
0002 025B 623 .WORD LNM PROCESS_LENGTH ; LNM$PROCESS STRING ITEM
0000015D' 025D 624 .WORD LNM$ STRING
00000000 025D 625 .ADDRESS LNM_PROCESS
00000000 0261 626 .LONG 0
0000 0265 627
0007 0265 628 .WORD LNM JOB LENGTH ; LNM$JOB STRING ITEM
0002 0267 629 .WORD LNM$ STRING
00000156' 0269 630 .ADDRESS LNM_JOB
00000000 026D 631 .LONG 0
00000000 0271 632
00000000 0271 633 .LONG 0 ; END OF ITEM LIST
0000 0275 634
000A 0275 635 LOG_S_LIST: ; ITEM LIST FOR LOG$SYSTEM
0002 0277 636 .WORD LNM SYSTEM LENGTH ; LNM$SYSTEM STRING ITEM
0000017D' 0279 637 .WORD LNM$ STRING
00000000 027D 638 .ADDRESS LNM_SYSTEM
00000000 0281 639 .LONG 0
00000000 0281 640
00000000 0281 641 .LONG 0 ; END OF ITEM LIST
0000 0285 642
0004 0285 643 SYSTEM_LIST: ; ITEM LIST FOR LNM$SYSTEM
0003 0285 644 .WORD 4 ; TERMINAL ATTRIBUTES ITEM
0003 0287 645 .WORD LNM$ ATTRIBUTES
```



```
00000129' 0289 646 .ADDRESS TERMINAL_BUFFER
00000000 028D 647 .LONG 0
          0291 648
          0010 0291 649 .WORD LNM_SYSTEM_TABLE_LENGTH ; LNM$SYSTEM_TABLE STRING ITEM
          0002 0293 650 .WORD LNM$_STRING
0000019B' 0295 651 .ADDRESS LNM_SYSTEM_TABLE
00000000 0299 652 .LONG 0
          029D 653
00000000 029D 654 .LONG 0 ; END OF ITEM LIST
          02A1 655
          0007 02A1 656 TEMPORARY_MAILBOX_LIST: ; ITEM LIST FOR LNM$TEMPORARY_MAILBOX
          0002 02A3 657 .WORD LNM_JOB_LENGTH ; LNM$JOB STRING ITEM
          00000156' 02A5 658 .WORD LNM$_STRING
00000000 02A9 659 .ADDRESS LNM_JOB
          02AD 660 .LONG 0
          02AD 661
00000000 02AD 662 .LONG 0 ; END OF ITEM LIST
          02B1 663
          0009 02B1 664 TRNLOG_GS_LIST: ; ITEM LIST FOR TRNLOG$_GROUP_SYSTEM
          0002 02B3 665 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
000001AB' 02B5 666 .WORD LNM$_STRING
00000000 02B9 667 .ADDRESS LOG_GROUP
          02BD 668 .LONG 0
          000A 02BD 669
          0002 02BF 670 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
000001BF' 02C1 671 .WORD LNM$_STRING
00000000 02C5 672 .ADDRESS LOG_SYSTEM
          02C9 673 .LONG 0
          02C9 674
00000000 02C9 675 .LONG 0 ; END OF ITEM LIST
          02CD 676
          000B 02CD 677 TRNLOG_PG_LIST: ; ITEM LIST FOR TRNLOG$_PROCESS_GROUP
          0002 02CF 678 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
000001B4' 02D1 679 .WORD LNM$_STRING
00000000 02D5 680 .ADDRESS LOG_PROCESS
          02D9 681 .LONG 0
          0009 02D9 682
          0002 02DB 683 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
000001AB' 02DD 684 .WORD LNM$_STRING
00000000 02E1 685 .ADDRESS LOG_GROUP
          02E5 686 .LONG 0
          02E5 687
00000000 02E5 688 .LONG 0 ; END OF ITEM LIST
          02E9 689
          000B 02E9 690 TRNLOG_PS_LIST: ; ITEM LIST FOR TRNLOG$_PROCESS_SYSTEM
          0002 02EB 691 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
000001B4' 02ED 692 .WORD LNM$_STRING
00000000 02F1 693 .ADDRESS LOG_PROCESS
          02F5 694 .LONG 0
          000A 02F5 695
          0002 02F7 696 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
000001BF' 02F9 697 .WORD LNM$_STRING
00000000 02FD 698 .ADDRESS LOG_SYSTEM
          0301 699 .LONG 0
          0301 700
00000000 0301 701 .LONG 0 ; END OF ITEM LIST
          0305 702
```

```
000B 0305 703 TRNLOG_PGS_LIST: ; ITEM LIST FOR TRNLOG$ PROCESS_GROUP_SYSTEM
0002 0305 704 .WORD LOG_PROCESS_LENGTH ; LOG$PROCESS STRING ITEM
00000184 0307 705 .WORD LNMS_STRING
00000000 0309 706 .ADDRESS LOG_PROCESS
030D 707 .LONG 0
0311 708
0009 0311 709 .WORD LOG_GROUP_LENGTH ; LOG$GROUP STRING ITEM
0002 0313 710 .WORD LNMS_STRING
000001AB 0315 711 .ADDRESS LOG_GROUP
00000000 0319 712 .LONG 0
031D 713
000A 031D 714 .WORD LOG_SYSTEM_LENGTH ; LOG$SYSTEM STRING ITEM
0002 031F 715 .WORD LNMS_STRING
000001BF 0321 716 .ADDRESS LOG_SYSTEM
00000000 0325 717 .LONG 0
0329 718
00000000 0329 719 .LONG 0 ; END OF ITEM LIST
032D 720
032D 721 :
032D 722 : ARGUMENT LISTS FOR THE $CRELNMS. THIS SYSTEM SERVICES CAN NOT BE DIRECTLY
032D 723 : ISSUED AT SYSTEM INITIALIZATION BECAUSE THE SWAPPER DOES NOT HAVE A P1 SPACE
032D 724 : WITH SYSTEM SERVICE VECTORS; HOWEVER, IT MAYBE CALLED DIRECTLY. SETUP AN
032D 725 : ARGUMENT LIST FOR EACH AND EVERY DIRECT CALL.
032D 726 :
032D 727
032D 728 DIRECTORIES_ARG: ; ARGUMENT LIST FOR LNMS$DIRECTORIES
032D 729 $CRELNM -
032D 730 ACMODE = KERNEL_MODE, -
032D 731 ATTR = LNM_NO_ALIAS, -
032D 732 ITMLST = DIRECTORIES_LIST, -
032D 733 LOGNAM = LNM_DIRECTORIES_DESC, -
032D 734 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0345 735
0345 736 FILE_DEV_EXEC_ARG: ; ARGUMENT LIST FOR EXECUTIVE LNMS$FILE_DEV
0345 737 $CRELNM -
0345 738 ACMODE = EXEC_MODE, -
0345 739 ITMLST = FILE_DEV_EXEC_LIST, -
0345 740 LOGNAM = LNM_FILE_DEV_DESC, -
0345 741 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
035D 742
035D 743 FILE_DEV_SUPER_ARG: ; ARGUMENT LIST FOR SUPERVISOR LNMS$FILE_DEV
035D 744 $CRELNM -
035D 745 ACMODE = SUPER_MODE, -
035D 746 ITMLST = FILE_DEV_SUPER_LIST, -
035D 747 LOGNAM = LNM_FILE_DEV_DESC, -
035D 748 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0375 749
0375 750 LOG_G_ARG: ; ARGUMENT LIST FOR LOG$GROUP
0375 751 $CRELNM -
0375 752 ACMODE = KERNEL_MODE, -
0375 753 ITMLST = LOG_G_LIST, -
0375 754 LOGNAM = LOG_G_DESC, -
0375 755 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
038D 756
038D 757 LOG_P_ARG: ; ARGUMENT LIST FOR LOG$PROCESS
038D 758 $CRELNM -
038D 759 ACMODE = KERNEL_MODE, -
```

```
0380 760 ITMLST = LOG_P_LIST, -
0380 761 LOGNAM = LOG_P_DESC, -
0380 762 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03A5 763
03A5 764 LOG_S_ARG: ; ARGUMENT LIST FOR LOG$SYSTEM
03A5 765 $CRELNM -
03A5 766 ACMODE = KERNEL_MODE, -
03A5 767 ITMLST = LOG_S_LIST, -
03A5 768 LOGNAM = LOG_S_DESC, -
03A5 769 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03BD 770
03BD 771 PERMANENT_MAILBOX_ARG: ; ARGUMENT LIST FOR LNM$PERMANENT_MAILBOX
03BD 772 $CRELNM -
03BD 773 ACMODE = KERNEL_MODE, -
03BD 774 ITMLST = PERMANENT_MAILBOX_LIST, -
03BD 775 LOGNAM = LNM_PERMANENT_MAILBOX_DESC, -
03BD 776 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03D5 777
03D5 778 SYSTEM_ARG: ; ARGUMENT LIST FOR LNM$SYSTEM
03D5 779 $CRELNM -
03D5 780 ACMODE = KERNEL_MODE, -
03D5 781 ATTR = LNM_NO_ALIAS, -
03D5 782 ITMLST = SYSTEM_LIST, -
03D5 783 LOGNAM = LNM_SYSTEM_DESC, -
03D5 784 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
03ED 785
03ED 786 TEMPORARY_MAILBOX_ARG: ; ARGUMENT LIST FOR LNM$TEMPORARY_MAILBOX
03ED 787 $CRELNM -
03ED 788 ACMODE = KERNEL_MODE, -
03ED 789 ITMLST = TEMPORARY_MAILBOX_LIST, -
03ED 790 LOGNAM = LNM_TEMPORARY_MAILBOX_DESC, -
03ED 791 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0405 792
0405 793 TRNLOG_GS_ARG: ; ARGUMENT LIST FOR TRNLOG$GROUP_SYSTEM
0405 794 $CRELNM -
0405 795 ACMODE = KERNEL_MODE, -
0405 796 ITMLST = TRNLOG_GS_LIST, -
0405 797 LOGNAM = TRNLOG_GS_DESC, -
0405 798 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
041D 799
041D 800 TRNLOG_PG_ARG: ; ARGUMENT LIST FOR TRNLOG$PROCESS_GROUP
041D 801 $CRELNM -
041D 802 ACMODE = KERNEL_MODE, -
041D 803 ITMLST = TRNLOG_PG_LIST, -
041D 804 LOGNAM = TRNLOG_PG_DESC, -
041D 805 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0435 806
0435 807 TRNLOG_PS_ARG: ; ARGUMENT LIST FOR TRNLOG$PROCESS_SYSTEM
0435 808 $CRELNM -
0435 809 ACMODE = KERNEL_MODE, -
0435 810 ITMLST = TRNLOG_PS_LIST, -
0435 811 LOGNAM = TRNLOG_PS_DESC, -
0435 812 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
044D 813
044D 814 TRNLOG_PGS_ARG: ; ARGUMENT LIST FOR TRNLOG$PROCESS_GROUP_SY
044D 815 $CRELNM -
044D 816 ACMODE = KERNEL_MODE, -
```



```
0440 817 ITMLST = TRNLOG_PGS_LIST, -
0440 818 LOGNAM = TRNLOG_PGS_DESC, -
0440 819 TABNAM = LNM_SYSTEM_DIRECTORY_DESC
0465 820
0000 0180 821 .PSECT $$$260 ; WRITABLE PSECT
0180 822 ; ITMLST MUST BE FOLLOWING TWO CRELNM
0180 823
0130 824 SYS_DISK_ARG: ; ARGUMENT LIST FOR SYSSDISK
0180 825 $CRELNM -
0180 826 ACMODE = EXEC_MODE, -
0180 827 LOGNAM = SYS_DISK_DESC, -
0180 828 TABNAM = LNM_SYSTEM_DESC
0198 829
0198 830 SYS_SYSDEVICE_ARG: ; ARGUMENT LIST FOR SYSSSYSDEVICE
0198 831 $CRELNM -
0198 832 ACMODE = EXEC_MODE, -
0198 833 LOGNAM = SYS_SYSDEVICE_DESC, -
0198 834 TABNAM = LNM_SYSTEM_DESC
```

```
01B0 836
00000465 837 .PSECT VF$LOWUSE ; PAGED PSECT AT END OF SYS.EXE
0465 838
0465 839
0465 840 :
0465 841 : DEFINE A QUOTA LIST TO BE USED BY VARIOUS PIECES OF THE SYSTEM WHEN
0465 842 : CREATING A SPECIAL SYSTEM PROCESS, LIKE A FILES-11 ACP. EVERY QUOTA
0465 843 : IS MENTIONED EXPLICITLY. NOTE THAT THIS LIST CAN BE TAILORED BY
0465 844 : COPYING IT TO SOME TEMPORARY LOCATION AND APPENDING NEW QUOTA ITEMS
0465 845 : TO THE END OF THE LIST. THE $CREPRC SYSTEM SERVICE USES THE LAST
0465 846 : VALUE OF A SPECIFIED QUOTA IN THE LIST WHEN IT CREATES A PROCESS.
0465 847 : NOTE THAT THE END OF THE LIST MUST BE TERMINATED BY A ZERO BYTE,
0465 848 : AND THAT THE LENGTH OF THE LIST, AS GIVEN BY PQL$C_SYSPQLLEN, DOES NOT
0465 849 : INCLUDE THE LIST TERMINATOR.
0465 850
0465 851 PQL$AB_SYSPQL::
01 0465 852 .BYTE PQL$_ASTLM ; SYSTEM PROCESS QUOTA LIST
0000000A 0466 853 .LONG 10 ; PROCESS AST LIMIT
02 046A 854 .BYTE PQL$_BIOLM ; PROCESS BUFFERED I/O LIMIT
0000000A 046B 855 .LONG 10
03 046F 856 .BYTE PQL$_BYTLM ; PROCESS BUFFERED I/O BYTE LIMIT
00008000 0470 857 .LONG 32768
04 0474 858 .BYTE PQL$_CPULM ; PROCESS CPU TIME LIMIT
00000000 0475 859 .LONG 0 ; ZERO IMPLIES NO LIMIT
05 0479 860 .BYTE PQL$_DIOLM ; PROCESS DIRECT I/O LIMIT
0000000A 047A 861 .LONG 10
06 047E 862 .BYTE PQL$_FILLM ; PROCESS OPEN FILE LIMIT
0000003C 047F 863 .LONG 60
07 0483 864 .BYTE PQL$_PGFLQUOTA ; PROCESS PAGE FILE QUOTA
00004E20 0484 865 .LONG 20000
08 0488 866 .BYTE PQL$_PRCLM ; PROCESS SUBPROCESS CREATION LIMIT
00000008 0489 867 .LONG 8
09 048D 868 .BYTE PQL$_TQELM ; PROCESS TIMER QUEUE ENTRY LIMIT
00000008 048E 869 .LONG 8
0B 0492 870 .BYTE PQL$_WSDEFAULT ; PROCESS DEFAULT WORKING SET SIZE
00000064 0493 871 .LONG 100
0A 0497 872 .BYTE PQL$_WSQUOTA ; PROCESS WORKING SET QUOTA
000000C8 0498 873 .LONG 200
0D 049C 874 .BYTE PQL$_WSEXTENT ; PROCESS WORKING SET EXTENT LIMIT
000003E8 049D 875 .LONG 1000
0C 04A1 876 .BYTE PQL$_ENQLM ; PROCESS LOCK LIMIT
00000064 04A2 877 .LONG 100
0E 04A6 878 .BYTE PQL$_JTQUOTA ; JOB-WIDE LOGICAL NAME TABLE QUOTA
00000400 04A7 879 .LONG 1024
00 04AB 880 10$: .BYTE PQL$_LISTEND ; END OF PROCESS QUOTA LIST
04AC 881
00000046 04AC 882 PQL$C_SYSPQLLEN == 10$ - PQL$AB_SYSPQL ; LENGTH OF LIST (MINUS TERMINATOR)
```

```
.SBTTL EXESSWAPINIT - INITIALIZATION AND STARTUP FOR SWAPPER

04AC 885
04AC 886
04AC 887
04AC 888
04AC 889
04AC 890
04AC 891
04AC 892
04AC 893
04AC 894
04AC 895
04AC 896
04AC 897
04AC 898
04AC 899
04AC 900
04B3 901
04B5 902
04BC 903
04BC 904
04BC 905
04BC 906
04BC 907
04BC 908
04BC 909
04C2 910
04C2 911
04C5 912
04CE 913
04D1 914
04D6 915
04D8 916
04DC 917
04E4 918
04E7 919
04E7 920
04E7 921
04E7 922
04EE 923
04F6 924
04FC 925
04FE 926
0505 927
0507 928
0507 929
050F 930
0512 931
0516 932
051A 933
051A 934
0521 935
0521 936
0521 937
0521 938
0521 939
0521 940
0521 941

      FUNCTIONAL DESCRIPTION:
      EXESSWAPINIT IS ENTERED WHEN THE SWAPPER PROCESS IS FIRST
      SCHEDULED AFTER A SYSTEM BOOT/STARTUP. THIS TRANSFER OCCURS
      VIA THE INITIAL PC VALUE BUILT INTO THE HARDWARE PCB FOR THE
      SWAPPER PROCESS. R4 CONTAINS THE ADDRESS OF THE SWAPPER PCB.

      EXESSWAPINIT::                                : SWAPPER INITIALIZATION
      INITIALIZE PAGED POOL.
      MOVL    G^EXESGL_PAGED,R11                      : POINT TO START OF PAGED POOL
      CLRL    (R11)+                                  : ZAP FORWARD LINK
      MOVL    G^SGNSGL_PAGEDYN,(R11)                  : AND SET SIZE

      ALLOCATE LOGICAL NAME HASH TABLE. THE NUMBER OF ENTRIES IN THE HASH TABLE
      MUST BE A POWER OF TWO. SO THE ALLOCATED SIZE IS THE SMALLEST POWER OF
      TWO LARGER THAN THE SYSGEN PARAMETER.
      PUSHL   @LNMSAL_HASHTBL                          : SAVE ADDR OF CRELNM ITMLST BLOCKS FOR
                                                         : "SYSSDISK" AND "SYSSSYSDEVICE"
      MOVL    #1,R8                                    : DO THIS TWICE
      40$:    SUBL3  #1,G^LNMSGL_HTBLSIZES[R8],R7       : PICK UP ONE LESS THAN SYSGEN PARAM
      CVTLF   R7,R7                                    : CONVERT TO FLOATING
      EXTZV   #7,#7,R7,R7                             : PICK UP EXPONENT-NOW THE POWER OF 2
      CLRL    R1                                       : CLEAR A REGISTER
      BBSS    R7,R1,50$                                : THE SIZE OF THE TABLE ROUNDED UP
      50$:    MOVL   R1,G^LNMSGL_HTBLSIZES[R8]          : WRITE BACK THE CORRECT VALUE
      SOBGEQ  R8,40$                                  : LOOP TWO TIMES

      INITIALIZE THE SYSTEM SPACE HASH TABLE.
      MOVL    G^LNMSGL_HTBLSIZES,R1                    : SIZE OF TABLE IN ENTRIES
      MOVAL   @LNMHSH$K_BUCKET[R1],R1                 : MULT BY 4 AND ADD OVERHEAD
      JSB     G^EXESALOPAGED                          : ALLOCATE MEMORY
      PUSHR   #^M<R1,R2>                               : SAVE REGISTERS DESTROYED BY MOVCS
      MOVCS   #0,#0,#0,R1,(R2)                       : ZERO HASH TABLE
      POPR    #^M<R1,R4>                               : RESTORE REGISTERS DESTROYED BY MOVCS
                                                         : NOTE: THAT R2 COMES BACK AS R4
      SUBL3   #1,G^LNMSGL_HTBLSIZES,R0                : CALC UPPER BOUND OF HASH INDEX
      MCOML   R0,LNMHSH$MASK(R4)                      : STORE HASH INDEX MASK
      MOVW    R1,LNMHSH$W_SIZE(R4)                   : STORE SIZE IN STRUCTURE HEADER
      MOVB    #DYN$C_RSHT,LNMHSH$B_TYPE(R4)          : STORE STRUCTURE TYPE
      MOVAB   (R4),@LNMSAL_HASHTBL                   : STUFF WAY POINTER TO TABLE
                                                         : NOTE: THAT THE HASH TABLE HAS BEEN
                                                         : INITIALIZED TO ZERO

      FIX UP THE SYSTEM LOGICAL NAME DIRECTORY, AND INSERT IT IN INTO THE
      APPROPRIATE HASH BUCKET OF THE SHAREABLE LOGICAL NAME HASH TABLE.
```



			9E	0521	942						
53	00000000'EF			0521	943	MOVAB	LNM\$SYSTEM_DIRECTORY,R3	:			
0000002C'EF	64	DE		0528	944	MOVAL	(R4),LNM_SYSTEM_DIR_LNMT+LNMT\$HASH	:	HASH TABLE ADDRESS IN LNMT		
				052F	945			:	: GET SIZE OF DIRECTORY NAME		
50	11 A3	9A		052F	946	MOVZBL	LNMBST_NAME(R3),R0	:			
51	12 A3	9E		0533	947	MOVAB	LNMBST_NAME+1(R3),R1	:	GET ADDRESS OF DIRECTORY NAME		
00000000'EF		16		0537	948	JSB	LNMSHASH	:	HASH THE DIRECTORY NAME		
	50 64	CA		053D	949	BICL2	LNMHSHSL_MASK(R4),R0	:	MODIFY THE INDEX TO BE IN RANGE		
OC A440	63	DE		0540	950	MOVAL	(R3),LNMRSHSC_BUCKET(R4)[R0]	:	INSERT DIRECTORY LNMB IN HASH TABLE		
				0545	951			:	LNMB\$LINK(R3)		
04 A3	OC A440	DE		0545	952	MOVAL	LNMRSHSC_BUCKET(R4)[R0],LNMB\$LINK(R3)	:	INSERT HASH TBL ENTRY IN DIRECT LNMB		
				054B	953			:			
				054B	954			:			
				054B	955			:			
				054B	956			:			
				054B	957			:			
				054B	958			:			
				054B	959			:			
51	000000C0'EF	9E		054B	960	MOVAB	SYSTEM_TABLE,R1	:	RETRIEVE SYSTEM TABLE LNMB ADDRESS		
00000000'FF		D0		0552	961	MOVL	@LNMSAC_HASH_TBL,-	:	MOVE THE ADDRESS OF THE SHAREABLE		
000000E8'EF				0558	962		SYSTEM_TABLE_LNMT+-	:	LOGICAL NAME HASH TABLE INTO THE		
				055D	963		LNMT\$HASH-	:	SYSTEM TABLE'S TABLE HEADER		
	52	D4		055D	964	CLRL	R2	:	NO SPECIAL INSERTION ATTRIBUTES		
00000000'GF	16			055F	965	JSB	G^LNMSINSLOGTAB	:	APPROPRIATELY INSERT LNM\$SYSTEM_TABLE		
				0565	966			:			
				0565	967			:			
				0565	968			:			
				0565	969			:			
				0565	970			:			
				0565	971			:			
				0565	972			:			
80000000'9F	FDC4 CF	FA		0565	973	CALLG -		:	CREATE LNM\$DIRECTORIES		
				056E	974		DIRECTORIES_ARG,-	:			
				056E	975		@#SYS\$CRELM-P1SYSVECTORS+^X80000000	:			
				056E	976			:			
80000000'9F	FDD3 CF	FA		056E	977	CALLG -		:	CREATE EXECUTIVE LNM\$FILE_DEV		
				0577	978		FILE_DEV_EXEC_ARG,-	:			
				0577	979		@#SYS\$CRELM-P1SYSVECTORS+^X80000000	:			
				0577	980			:			
80000000'9F	FDE2 CF	FA		0577	981	CALLG -		:	CREATE SUPERVISOR LNM\$FILE_DEV		
				0580	982		FILE_DEV_SUPER_ARG,-	:			
				0580	983		@#SYS\$CRELM-PTSYSVECTORS+^X80000000	:			
				0580	984			:			
80000000'9F	FDF1 CF	FA		0580	985	CALLG -		:	CREATE LOG\$GROUP		
				0589	986		LOG_G_ARG,-	:			
				0589	987		@#SYS\$CRELM-P1SYSVECTORS+^X80000000	:			
				0589	988			:			
80000000'9F	FE00 CF	FA		0589	989	CALLG -		:	CREATE LOG\$PROCESS		
				0592	990		LOG_P_ARG,-	:			
				0592	991		@#SYS\$CRELM-P1SYSVECTORS+^X80000000	:			
				0592	992			:			
80000000'9F	FE0F CF	FA		0592	993	CALLG -		:	CREATE LOG\$SYSTEM		
				0598	994		LOG_S_ARG,-	:			
				0598	995		@#SYS\$CRELM-P				

```
80000000'9F  FE2D CF  FA  05A4  999
                        05A4  1000
                        05A4  1001
                        05AD  1002
                        05AD  1003
80000000'9F  FE3C CF  FA  05AD  1004
                        05AD  1005
                        05B6  1006
                        05B6  1007
80000000'9F  FE4B CF  FA  05B6  1008
                        05B6  1009
                        05BF  1010
                        05BF  1011
80000000'9F  FE5A CF  FA  05BF  1012
                        05BF  1013
                        05C8  1014
                        05C8  1015
80000000'9F  FE69 CF  FA  05C8  1016
                        05C8  1017
                        05D1  1018
                        05D1  1019
80000000'9F  FE78 CF  FA  05D1  1020
                        05D1  1021
                        05DA  1022
                        05DA  1023
                        05DA  1024
                        05DA  1025
                        05DA  1026
                        05DA  1027
                        05DA  1028
00000194'EF  54 8ED0  05DA  1029
                        64 DE  05DD  1030
000001AC'EF  0000'C4 DE  05E4  1031
                        05E4  1032
                        05ED  1033
                        05ED  1034
80000000'9F  00000180'EF FA  05ED  1035
                        05F8  1036
                        05F8  1037
                        05F8  1038
80000000'9F  00000198'EF FA  05F8  1039
                        0603  1040
                        0603  1041
                        0603  1042
                        51 00'8F 9A 0603 1043
                        50 54 D0 0607 1044
00000000'EF  16 060A 1045
                        0610 1046
                        0610 1047
                        0610 1048
                        0610 1049
                        0610 1050
                        0610 1051
                        0610 1052
                        0610 1053
                        0610 1054
                        0610 1055

CALLG -                ; CREATE LNM$SYSTEM
SYSTEM ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE LNM$TEMPORARY_MAILBOX
TEMPORARY_MAILBOX_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE TRNLOG$_GROUP_SYSTEM
TRNLOG GS_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE TRNLOG$_PROCESS_GROUP
TRNLOG PG_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE TRNLOG$_PROCESS_SYSTEM
TRNLOG PS_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE TRNLOG$_PROCESS_GROUP_SYSTEM
TRNLOG PGS_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

::
:: CREATE TWO STARTUP LOGICAL NAMES.
::
::
POPL R4                ; RECOVER ADDR OF CRELOG BLOCK
MOVAL (R4),SYS_DISK_ARG+CRELNM$_ITMLST
:: STUFF THE ADDRESS OF THE ITEM LIST
MOVAL BDL$_SYSDLOG(R4),SYS_SYSD_DEVICE_ARG+CRELNM$_ITMLST
:: STUFF THE ADDRESS OF THE ITEM LIST

CALLG -                ; CREATE SYS$DISK
SYS_DISK_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

CALLG -                ; CREATE SYS$SYSDEVICE
SYS_SYSD_DEVICE_ARG, -
@#SYS$CRELNM-P1SYSVECTORS+^X80000000

MOVZBL #BDL$_CRELNM_ITMLST,R1 ; GET THE SIZE OF THE STRUCTURE
MOVL R4,R0                ; MOVE STRUCTURE ADDR INTO CORRECT REG
JSB EXE$DEANONPGDSIZ      ; RETURN THE MEMORY

::
:: CREATE INITIAL PROCESSES
::
:: THE $CREPRC S MACRO CANNOT BE USED BECAUSE THAT MACRO GENERATES A
:: CALL THROUGH THE P1 SYSTEM SERVICE VECTOR PAGES AND THE SWAPPER DOES
:: NOT HAVE A P1 SPACE. THE SENSE OF THE CREATE PROCESS CALL IS THE
:: FOLLOWING.
$CREPRC_S INPUT=TTODESC,- ;
```

			0610	1056	:	OUTPUT=TTODESC,-;
			0610	1057	:	ERROR=TTODESC,-;
			0610	1058	:	IMAGE=IMGDESC,-;
			0610	1059	:	UIC=#^X80020,-;
			0610	1060	:	STSFLG=#<PRCSM_NOACNT!PRCSM_SSRWAIT>,-
			0610	1061	:	BASPRI=#2
			0610	1062	:	
	09	DD	0610	1063		PUSHL #<PRCSM_NOACNT!PRCSM_SSRWAIT>
	7E	D4	0612	1064		CLRL -(SP)
00080020	8F	DD	0614	1065		PUSHL #^X80020
	02	DD	061A	1066		PUSHL #2
	7E	7C	061C	1067		CLRQ -(SP)
	00	DD	061E	1068		PUSHL #0
F9EF	CF	7F	0620	1069		PUSHAQ TTODESC
	6E	DD	0624	1070		PUSHL (SP)
	6E	DD	0626	1071		PUSHL (SP)
F9D4	CF	7F	0628	1072		PUSHAQ IMGDESC
	00	DD	062C	1073		PUSHL #0
80000000'9F	0C	FB	062E	1074		CALLS #12,#SYS\$CREPRC-P1SYSVECTORS+^X80000000
00000000'GF	17	0635	1075			JMP G^LOOP ; JUMP OFF TO THE MAIN LOOP



```
0638 1078 .SBTTL SWAPPER - MAIN LOOP
0638 1079
0638 1080 :++
0638 1081 : FUNCTIONAL DESCRIPTION:
0638 1082 : THE MAIN LOOP OF THE SWAPPER IS EXECUTED WHENEVER THE SWAPPER IS AWAKENED
0638 1083 : FOR ANY REASON. EACH OF THE FUNCTIONAL ROUTINES WILL CHECK TO SEE IF
0638 1084 : THEY HAVE ANY ACTION TO PERFORM.
0638 1085 :--
0638 1086
00000000 1087
2E 10 0000 1088 LOOP: .PSECT $AEXENONPAGED : NON-PAGED PSECT
FFFB' 30 0002 1089 BSBW BALANCE : BALANCE FREE PAGE COUNT
0088 30 0005 1090 BSBW MMGSWRTMFYPAG : WRITE MODIFIED PAGES
0000'CF 05 0008 1091 BSBW SWAPSCHEG : SCHEDULE SWAP
06 13 000C 1092 TSTL W*EXESGL_PFATIM : CHECK FOR POWER FAIL TIME
00000000'EF 16 000E 1093 BEQL 15$ : BRANCH IF NO POWERFAIL
54 0000'CF 00 0014 1094 JSB EXESPOWERAST : GIVE ANY REQUIRED POWER FAIL ASTS
52 0000'CF 00 0019 1095 MOVL W*SCH$GL_CURPCB,R4 : GET PROPER PCB ADDRESS
05 24 A4 0C E4 0021 1096 MOVAQ W*SCH$GQ_HIBWQ,R2 : AND ADDRESS OF WAIT QUEUE HEADER
00 DD 0026 1097 SETIPL #IPL$ SYNCH : BLOCK SYSTEM EVENTS WHILE CHECKING
FFD5' 30 0028 1098 BBSC #PCB$V_WAKEPEN,PCB$S_STS(R4),20$ : TEST AND CLEAR WAKE PENDING
00 002B 1100 20$: PUSHL #0 : NULL PSL
DO 11 002E 1101 BSBW SCH$WAITK : WAIT WITH STACK CLEAN
0030 1102 SETIPL #0 : DROP IPL
BRB LOOP : CHECK FOR WORK TO DO
.DISABLE LSB
```

```
0030 1105 .SBTTL BALANCE FREE PAGE COUNT
0030 1106
0030 1107
0030 1108
0030 1109
0030 1110
0030 1111
0030 1112
0030 1113
0030 1114
0030 1115
0030 1116
0030 1117 BALANCE:
53 0000'CF 0000'CF 0A 15 0037 1118 CMPL W^SGN$GL_FREELIM,W^SCH$GL_FREECNT ; ARE WE HERE DUE TO FREELIM?
0000'CF 0000'CF 0B 19 0039 1119 SUBL3 W^SGN$GL_FREEGOAL,W^SCH$GL_FREECNT,R3 ; SUFFICIENT FREE PAGES?
0000'CF 0B 19 0041 1120 BLSS 20$ ; NO, MUST ACQUIRE SOME
04 13 0043 1121 5$: TSTW W^SCH$GW_DELPHDCT ; CHECK FOR DELETED PROCESS HEADERS
53 13 0047 1122 BEQL 10$ ; NONE, EXIT
1E 11 0049 1123 CLRL R3 ; INDICATE NO FREE PAGES NEEDED
05 004B 1124 BRB 25$
004D 1125 10$:
004E 1126 20$:
50 17 0000'CF 00' E0 004E 1127 BBS S^#SCH$V_MPW,W^SCH$GB_SIP,25$ ; MODIFIED PAGE WRITING ACTIVE
0000'CF 0000'CF 0D 15 0054 1128 SUBL3 W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYCNT,R0 ; HOW MUCH WILL WRITING PAGES
50 53 005C 1129 BLEQ 25$ ; NONE, MUST OUTSWAP
0000'CF 0000'CF 08 19 005E 1130 ADDL R3,R0 ; YIELD RELATIVE TO WHAT WE NEED?
0000'CF 0000'CF 08 19 0061 1131 BLSS 25$ ; NOT ENOUGH, MUST OUTSWAP
0000'CF 0000'CF 0D 05 0063 1132 MOVL W^SCH$GL_MFYLOLIM,W^SCH$GL_MFYLM ; TRIGGER MODIFIED PAGE WRITING
006A 1133 RSB ; AND EXIT TO LET IT HAPPEN
006B 1134
006B 1135
006B 1136
006B 1137
006B 1138
006B 1139
006B 1140 25$:
DC 0000'CF 00' E2 006B 1140 BBSS S^#SCH$V_SIP,W^SCH$GB_SIP,10$ ; EXIT IF SWAPPER ALREADY BUSY
3FC0 8F BB 0071 1141 SETIPL #IPL$ SYNCH ; BLOCK SYSTEM EVENTS
0000'CF 94 0074 1142 PUSHF #M<R5,R7,R8,R9,R10,R11,AP,FP> ; SAVE NON-STANDARD REGISTERS
5D 53 0078 1143 CLRB W^SWP$GB_ISWPRI ; SET PRIORITY FOR SWAP SCHEDULE
5D 53 007C 1144 MOVL R3,FP ; GET AND TEST FREE PAGE DEFICIT
0018'CF 08 18 007F 1145 BGEQ 30$ ; NONE, PURGING DELETED HEADERS
0018'CF 02 12 0081 1146 TSTW W^SWP$GW_BALCNT ; CHECK FOR SINGULAR BALANCE SET
5D 04 0085 1147 BNEQ 30$ ; NO, CAN OUTSWAP
008A 31 0087 1148 CLRL FP ; PREVENT OUTSWAP SCHEDULE
0089 1149 30$: BRW OUTSWAP ; TRY TO FORCE AN OUTSWAP
```

FUNCTIONAL DESCRIPTION:  
BALANCE WILL ENSURE THAT THE FREE PAGE LIST HAS AT LEAST THE NUMBER OF  
PAGES SPECIFIED BY THE PARAMETER FREELIM. IF NOT, PAGES WILL BE MADE  
AVAILABLE BY EITHER WRITING MODIFIED PAGES OR OUTSWAPPING PROCESSES.  
IF SUFFICIENT FREE PAGES ARE AVAILABLE, THEN A CHECK IS MADE FOR  
DELETED PROCESS HEADERS IN NEED OF CLEANUP.

```
008C 1152 .SBTTL SCHEDULE SWAP
008C 1153
008C 1154 :++
008C 1155 : FUNCTIONAL DESCRIPTION:
008C 1156 : SWAPSCHEG IS CALLED BY THE MAIN LOOP OF THE SWAPPER PROCESS TO CHECK
008C 1157 : ELIGIBLE INSWAP CANDIDATES AND TO PROVIDE MEMORY NEEDED FOR THEIR
008C 1158 : INSWAP. A QUICK EXIT WILL BE TAKEN IF THE SWAPPER IS ALREADY BUSY.
008C 1159 : NO OUTSWAP WILL BE NEEDED IF THE NUMBER OF REQUIRED PAGES CAN BE
008C 1160 : TAKEN FROM THE FREE PAGE LIST LEAVING AT LEAST FREELIM STILL FREE.
008C 1161 : OTHERWISE OUTSWAP WILL BE ENTERED TO MAKE PAGES AVAILABLE BY ANY
008C 1162 : MEANS NECESSARY.
008C 1163 :--
008C 1164
008C 1165 QEMPTY: BUG_CHECK QUEUEMPTY,FATAL ; EMPTY QUEUE OR NOT A PCB
0090 1166
0090 1167 SWAPSCHEG:
0090 1168 SETIPL #IPL$ SYNCH ; BLOCK SYSTEM EVENTS
0093 1169 BBSS S^#SCH$V_SIP,W^SCH$GB_SIP,5$ ; EXIT IF SWAP IN PROGRESS
0099 1170 FFS #0,#32,W^SCH$GL_COMOQS,R2 ; FIND HIGHEST PRIORITY QUEUE
00A0 1171 BNEQ 10$ ; FOUND ONE
00A2 1172 BBCC S^#SCH$V_SIP,W^SCH$GB_SIP,5$ ; CLEAR SWAP IN PROGRESS
00A8 1173 5$: SETIPL #0 ; DROP IPL
00AB 1174 RSB ; AND RETURN
00AC 1175
00AC 1176 10$:
00AC 1177 PUSHF #M<R6,R7,R8,R9,R10,R11,AP,FP> ; SAVE REGS OTHER THAN R0-R5
00B0 1178 MOVAQ W^SCH$AQ_COMOH[R2],R3 ; COMPUTE ADDRESS OF QUEUE HEADER
00B6 1179 MOVL (R3),R4 ; GET PCB ADDRESS
00B9 1180
00B9 1181 : THE FOLLOWING CHECK IS NEEDED DUE TO THE ODIOUS MISLEADING SYMPTOMS THAT
00B9 1182 : MIGHT OTHERWISE RESULT.
00B9 1183
00B9 1184 CMPB #DYN$C_PCB,PCB$B_TYPE(R4); IS THIS A GOOD PCB?
00BD 1185 BNEQ QEMPTY ; BUGCHECK IF NOT
00BF 1186
00BF 1187 : DETERMINE THE SIZE OF THE INSWAP CANDIDATE, TAKING INTO ACCOUNT THE FACT
00BF 1188 : THAT THE PROCESS HEADER MIGHT ALREADY BE RESIDENT.
00BF 1189
00BF 1190 MOVZWL PCB$W_PPGCNT(R4),R0 ; COUNT OF PROCESS PAGES
00C3 1191 MOVZWL PCB$W_GPGCNT(R4),R10 ; COUNT OF GLOBAL PAGES
00C7 1192 ADDL R0,R10 ; SUM PAGE COUNTS
00CA 1193 BBC #PCB$V_PHDRES,PCB$S_STS(R4),15$ ; CONTINUE IF HEADER NON-RESIDENT
00CF 1194 MOVZWL PCB$W_APTCNT(R4),R0 ; GET ACTIVE PAGE TABLE COUNT
00D3 1195 SUBL R0,R10 ; SUBTRACT RESIDENT HEADER PAGES FROM REQUIR
00D6 1196 15$:
00D6 1197 SUBL3 W^SCH$GL_FREELIM,W^SCH$GL_FREECNT,R0 ; COMPUTE PAGES AVAILABLE
00DE 1198 MOVB PCB$B_PRT(R4),W^SWP$GB_ISWPRI ; SAVE PRIORITY OF INSWAP
00E4 1199 SUBL3 R10,R0,FP ; WILL PROCESS FIT?
00E8 1200 BLSS 20$ ; NO, MUST OUTSWAP
00EA 1201 MOVW W^SCH$GW_SWPFAIL,W^SCH$GW_SWPFCNT ; RESET FAILURE COUNTER
00F1 1202 BRW INSWAP ; YES PERFORM SWAP
00F4 1203
00F4 1204 : IF INSWAPPING A NON-REAL TIME PROCESS, THEN CHECK TO SEE IF ITS CURRENT
00F4 1205 : PRIORITY IS THE DEFAULT BACKGROUND PRIORITY. IF SO, THEN DELAY AT LEAST
00F4 1206 : SWAPRATE INTERVAL SINCE THE LAST INSWAP. THE EFFECT WILL BE TO AVOID FILLING
00F4 1207 : THE BALANCE SET WITH CRUNCHING PROCESSES IMMEDIATELY.
00F4 1208 :
```

52 OF 0000'CF 00' E2 0093 1169  
0000'CF 20 00 EA 0099 1170  
00 0000'CF 00' E5 00A0 1171  
00A2 1172  
00A8 1173 5\$:  
00AB 1174  
00AC 1175  
00AC 1176 10\$:  
00AC 1177  
00B0 1178  
00B6 1179  
00B9 1180  
00B9 1181  
00B9 1182  
00B9 1183  
00B9 1184  
00BD 1185  
00BF 1186  
00BF 1187  
00BF 1188  
00BF 1189  
00BF 1190  
00C3 1191  
00C7 1192  
00CA 1193  
00CF 1194  
00D3 1195  
00D6 1196 15\$:  
00D6 1197  
00DE 1198  
00E4 1199  
00E8 1200  
00EA 1201  
00F1 1202  
00F4 1203  
00F4 1204  
00F4 1205  
00F4 1206  
00F4 1207  
00F4 1208

53 3FC0 BF BB 00AC 1177  
0000'CF42 7E 00B0 1178  
54 63 D0 00B6 1179  
00B9 1180  
00B9 1181  
00B9 1182  
00B9 1183  
00B9 1184  
00BD 1185  
00BF 1186  
00BF 1187  
00BF 1188  
00BF 1189  
00BF 1190  
00C3 1191  
00C7 1192  
00CA 1193  
00CF 1194  
00D3 1195  
00D6 1196  
00D6 1197  
00DE 1198  
00E4 1199  
00E8 1200  
00EA 1201  
00F1 1202  
00F4 1203  
00F4 1204  
00F4 1205  
00F4 1206  
00F4 1207  
00F4 1208

50 36 A4 3C 00BF 1190  
5A 34 A4 3C 00C3 1191  
07 24 A4 12 E1 00CA 1193  
50 30 A4 3C 00CF 1194  
5A 50 C2 00D3 1195  
00D6 1196  
00D6 1197  
00DE 1198  
00E4 1199  
00E8 1200  
00EA 1201  
00F1 1202  
00F4 1203  
00F4 1204  
00F4 1205  
00F4 1206  
00F4 1207  
00F4 1208

50 0000'CF 0000'CF C3 00D6 1197  
0000'CF 0B A4 90 00DE 1198  
5D 50 5A C3 00E4 1199  
0A 19 00E8 1200  
001A'CF 0000'CF B0 00EA 1201  
0441 31 00F1 1202  
00F4 1203  
00F4 1204  
00F4 1205  
00F4 1206  
00F4 1207  
00F4 1208



```
          OB A4 10 91 00F4 1209 20$:
          1C 14 00F4 1210
51 1F 0000'CF 83 00F8 1211
          51 52 91 00FA 1212
          11 19 0100 1213
          51 00000000'EF D0 0103 1214
          0000'CF 51 D1 0105 1215
          03 1A 010C 1216
          08CB 31 0111 1217
          0113 1218
          0116 1219 40$:
```

```
CMPB #16,PCB$B_PRI(R4)
BGTR 40$
SUBB3 W^SY$GB_DEFPRI,#31,R1
CMPB R2,R1
BLSS 40$
MOVL EX$GQ_SYSTIME,R1
CMPL R1,W^SWP$GL_SWTIME
BGTRU 40$
BRW SWAPEXIT
```

```
: SCHEDULE OUTSWAP
: IS THIS A REAL TIME PROCESS?
: BR IF SO
: CONVERT PRIORITY TO INTERNAL FORM
: IS THIS A CRUNCHER OR LOW PRIORITY JOB?
: BR IF NOT
: GET CURRENT TIME IN APPROX. 10MS UNITS
: HAS INTERVAL ELAPSED?
: BR IF YES
: CAN'T DO SWAP NOW
:
```

```
0116 1222 .SBTTL OUTSWAP
0116 1223
0116 1224
0116 1225
0116 1226
0116 1227
0116 1228
0116 1229
0116 1230
0116 1231
0116 1232
0116 1233
0116 1234
0116 1235
0116 1236
0116 1237
0116 1238
0116 1239
0116 1240
0116 1241
0116 1242
0116 1243
0116 1244
0116 1245
0116 1246
0116 1247
0116 1248
0116 1249
0116 1250
0116 1251
0116 1252
0116 1253
0116 1254
0116 1255
0116 1256
0116 1257
0116 1258
0116 1259
0116 1260
0116 1261
0116 1262
0116 1263
0116 1264
0116 1265
0116 1266
0116 1267
0116 1268
0116 1269
0116 1270
0116 1271
0116 1272
0116 1273
0116 1274
0116 1275
0116 1276
0116 1277
0116 1278
```

SCHEDULE AND PERFORM OUTSWAPS IF POSSIBLE

FUNCTIONAL DESCRIPTION:  
THE OUTSWAP STRATEGY IS TO FREE PROCESS HEADERS FOR OUTSWAP PROCESSES,  
USE AVAILABLE MODIFIED PAGES (AFTER WRITING THEM) AND FINALLY AS A LAST  
RESORT OUTSWAP ANOTHER PROCESS. ONLY ONE OF THESE ACTIONS WILL BE TAKEN  
AT A TIME THEN THE SCHEDULING SITUATION WILL BE RE-EVALUATED. THE VALUE  
IN FP INDICATES THE SIZE OF THE PAGE DEFICIT AND WILL BE SET POSITIVE IF  
ENTERED FROM BALANCE TO FREE DELETED PROCESS HEADERS.

INPUT: FP - NEGATIVE VALUE WILL PERMIT PROCESS OUTSWAP  
ZERO OR POSITIVE WILL PURGE HEADERS ONLY.

OUTSWAP:

```
58 00000000'EF 01 C3 0116 1242 SUBL3 #1,SGN$GL_BALSETCT,R8 ; TRY TO OUTSWAP
59 00  D2 0116 1243 MCOML #0,R9 ; INIT INDEX FOR BALANCE SLOT SCAN
0000'DF48 03 B5 0121 1244 10$: TSTW @W'PHV$GL_REFCBAS[R8] ; INDICATE NO FREE LIST PURGE CANDIDATE
00D1 31 0126 1245 BNEQ 12$ ; IS SLOT IN NEED OF CLEANUP?
54 0000'DF48 32 0128 1246 BRW 60$ ; CONTINUE IF NOT RELEASABLE
0A 15 0131 1247 12$: CVTWL @W'PHV$GL_PIXBAS[R8],R4 ; GO RELEASE PAGE TABLES AND HEADER
54 0000'DF44 D0 0133 1248 BLEQ 15$ ; GET PROCESS INDEX
OF 24 A4 E8 0139 1249 MOVL @W'SCH$GL_PCBVEC[R4],R4 ; DELETED PROCESS OR VACANT SLOT
0D 13 013D 1250 BLBS PCB$STST(R4),20$ ; GET PCB ADDRESS FOR PIX
0000'CF 95 013F 1251 15$: BEQL 20$ ; SKIP IF PROCESS IS RESIDENT
04 13 0143 1252 TSTB W'EXE$GQ_SYSTIME ; VACANT SLOT
59 03 0145 1253 BEQL 17$ ; ADD 1 IN 8 RANDOMNESS TO DECISION
59 03 0147 1254 TSTL R9 ; BRANCH ON LOW PROBABILITY
D2 58 D0 0149 1255 BGEQ 20$ ; CHECK FOR REMEMBERED INDEX
0001'CF 95 014F 1256 17$: MOVL R8,R9 ; YES DONT OVERWRITE
05 13 0153 1257 20$: SOBGEQ R8,10$ ; SAVE BALANCE SLOT NUMBER OF CANDIDATE
58 59 D0 0155 1258 TSTB W'EXE$GQ_SYSTIME+1 ; TRY ALL SLOTS
11 18 0158 1259 BEQL 21$ ; ADD 1 IN 256 RANDOMNESS TO DECISION
5D D5 015A 1260 MOVL R9,R8 ; BRANCH ON VERY LOW PROBABILITY
07 18 015C 1261 BGEQ 24$ ; GET AND TEST SLOT INDEX FOR SECONDARY CAND
21$: TSTL FP ; BR IF SLOT FOR CLEANUP
BGEQ 22$ ; CHECK FOR HEADER PURGE
; EXIT IF SO

; SINCE THERE WAS NO HEADER TO FREE, WE MUST NOW WRITE MODIFIED PAGES OR OUTSWAP
; SOME PROCESS. ONLY IF MODIFIED PAGES (MFYCNT-LOLIM) WILL TOTALLY SATISFY OUR
; NEEDS WILL THEY BE WRITTEN. OTHERWISE THE LEAST USEFUL (BY SOME OPINION) PROCESS
; WILL BE OUTSWAPPED AND THE SITUATION RECONSIDERED.

FE9F' 30 015E 1270 BSBW SCH$OSWPSCHED ; SCHEDULE OUTSWAP
54 D5 0161 1271 TSTL R4 ; ANY CANDIDATE?
03 12 0163 1272 BNEQ 23$ ; YES
0879 31 0165 1273 22$: BRW SWAPEXIT ; ELSE EXIT AND TRY LATER
0094 31 0168 1274 23$: BRW 70$ ; GO DO OUTSWAP

; A HEADER SLOT IN NEED OF CLEANUP WAS FOUND, NOW SCAN THE FREELIST FOR ALL
; PAGES WHOSE PTE BACK POINTERS PLACE THEM WITHIN THIS HEADER. DELETE THE
; CONTENT OF THOSE PAGES VIA MMG$DELCONPFN TO FINALLY REDUCE THE REFERENCE
```

```
016B 1279 : COUNT BINDING THE HEADER.
016B 1280 :
57 0000'CF 02 9C 016B 1281 24$: ROTL #2,W*SWP$GL_BSL0TSZ,R7 : GET SIZE OF BALANCE SLOT IN BYTES
56 58 57 C5 0171 1282 MULL3 R7,R8,R6 : COMPUTE OFFSET TO BASE OF SLOT
0000'DF46 9F 0175 1283 PUSHAB @W*SWP$GL_BALSPT[R6] : ADD BASE TO GET ADDRESS
52 56 07 9C 017A 1284 ROTL #7,R6,R2 : FORM OFFSET TO PHD BASE
0000'DF42 9F 017E 1285 PUSHAB @W*SWP$GL_BALBASE[R2] : BASE ADDRESS FOR PHD
51 57 07 9C 0183 1286 ROTL #7,R7,R1 : MUL SPT SLOT SIZE BY 128
04 BE47 9F 0187 1287 PUSHAB @4(SP)[R7] : FORM HIGH LIMIT FOR PAGTBLPTE
04 BE41 9F 018B 1288 PUSHAB @4(SP)[R1] : ANS SAVE PTE HIGH LIMIT
018F 1289 :
018F 1290 :
018F 1291 :
018F 1292 :
018F 1293 :
018F 1294 :
018F 1295 :
018F 1296 :
018F 1297 :
018F 1298 :
018F 1299 :
018F 1300 :
57 D4 018F 1301 CLRL R7 : ASSUME ONLY FREELIST SCAN
0000'DF48 B5 0191 1302 TSTW @W*PHV$GL_PIXBAS[R8] : IS THIS A DELETED PROCESS HEADER?
03 18 0196 1303 BGEQ 25$ : BR IF NOT. ONLY SCAN FREELIST
57 02 D0 0198 1304 MOVL #PFNSC_BADPAGLST,R7 : INITIALIZE LOOP SCAN TO BADPAGLST
50 0000'CF47 D0 019B 1305 25$: MOVL W*PFNS$L_HEAD[R7],R0 : GET HEAD OF LIST TO START SCAN
3E 13 01A1 1306 BEQL 45$ : NO PAGES, DONE
01A3 1307 30$: PFN REFERENCE -
01A3 1308 MOVZWL <@W*PFNS$Lx FLINK[R0],R9>,- : GET FORWARD LINK
01A3 1309 LONG_OPCODE=MOVL,-
01A3 1310 IMAGE=SYS_NONPAGED
53 0000'DF40 D0 01A9 1311 MOVL @W*PFNS$L_PTE[R0],R3 : GET SVA OF PTE FOR PAGE
01AF 1312 ASSUME PFNSC_PPGTBL EQ 4
01AF 1313 ASSUME PFNSC_GPGTBL EQ 5
51 0000'DF40 01 02 EF 01AF 1314 EXTZV #2,#1,@W*PFNS$L_TYPE[R0],R1 : GET PAGE TABLE BIT
08 AE41 53 D1 01B7 1315 CMPL R3,B(SP)[R1] : COMPARE WITH LOW LIMIT
1E 1F 01BC 1316 BLSSU 40$ : OUT OF RANGE
6E41 53 D1 01BE 1317 CMPL R3,(SP)[R1] : COMPARE WITH HIGH LIMIT
18 1E 01C2 1318 BGEQU 40$ : OUT OF RANGE
OC 57 E9 01C4 1319 BLBC R7,35$ : BR IF FREE OR BAD LIST
52 00000000'FF40 D0 01C7 1320 MOVL @PFNS$L_BAK[R0],R2 : GET BACKING STORE ADDRESS
09 52 16 E0 01CF 1321 BBS #PTESV_TYPO,R2,40$ : LEAVE MODIFIED SECTION PAGES
52 57 D0 01D3 1322 35$: MOVL R7,R2 : SET LIST NUMBER FOR DELETE
FE27' 30 01D6 1323 BSBW MMGSREMPFN : REMOVE PAGE FROM FREE LIST
0895 30 01D9 1324 BSBW RELDELPAGE : RELEASE PAGE DELETING CONTENT
50 59 D0 01DC 1325 40$: MOVL R9,R0 : FLINK TO NEXT PAGE
C2 12 01DF 1326 BNEQ 30$ : ANOTHER PAGE TO TRY
B7 57 F4 01E1 1327 45$: SOBGEQ R7,25$ : NEXT LIST
5E 10 C0 01E4 1328 ADDL #16,SP : CLEAN STACK OF LIMITS
0000'DF48 B5 01E7 1329 TSTW @W*PHV$GL_REF(BAS[R8]) : DID WE FREE PROCESS HEADER
OE 13 01EC 1330 BEQL 60$ : YES, RELEASE IT
01EE 1331 :
01EE 1332 : THERE ARE TWO REASONS THAT MIGHT PREVENT THE HEADER FROM BEING RELEASED BY
01EE 1333 : THE FREELIST SCAN:
01EE 1334 :
01EE 1335 :
```

1. SOME OF THE TRANSITION PAGES ARE ON THE MODIFIED LIST.  
2. THERE IS I/O IN PROGRESS ON THE TRANSITION PAGES.



```
01EE 1336 : TO COVER THE FORMER CASE (SINCE WE CANT REALLY TELL), THE MODIFIED LIST MUST
01EE 1337 : BE TOTALLY FLUSHED. HOWEVER THIS IS ACTUALLY QUITE RARE.
01EE 1338 :
0000'CF D4 01EE 1339 : CLRL W^SCH$GL_MFYLOLIM : FORCE ENTIRE MODIFY LIST TO BE WRITTEN
0000'CF B4 01F2 1340 : CLRW W^SCH$GL_MFYLIM : CLEAR PART OF HI LIMIT, NOT PART THAT
01F6 1341 : : INDICATES MODIFIED WRITING IN PROGRESS
59 00 D2 01F6 1342 : MCOML #0,R9 : NO, TRY FOR ANOTHER
FF50 31 01F9 1343 : BRW 20$ : NOW ATTEMPT CLEANUP
0117 31 01FC 1344 60$: BRW RELPHD : GO RELEASE PROCESS HEADER
01FF 1345 70$:
01FF 1346 :
01FF 1347 :
01FF 1348 :
01FF 1349 :
55 6C A4 D0 01FF 1350 : MOVL PCB$L_PHD(R4),R5 : GET PROCESS HEADER ADDRESS
0203 1351 :
0018'CF B7 0203 1352 : DECW W^SWP$GW_BALCNT : DECREASE NUMBER IN BALANCE SET
0857 30 0207 1353 : BSBW OSINIT : INIT REGISTERS FOR SCAN
30 A4 B4 020A 1354 : CLRW PCB$W_APTCNT(R4) : INITIALIZE ACTIVE PAGE TABLE COUNT
57 08 A5 3C 020D 1355 : MOVZWL PHD$W_WSLIST(R5),R7 : WS INDEX FOR PERM PAGES
56 12 A5 3C 0211 1356 : MOVZWL PHD$W_WSLAST(R5),R6 : END OF WORKING SET LIST
0215 1357 :
0215 1358 :
0215 1359 :
0215 1360 :
0215 1361 :
0215 1362 :
0215 1363 :
0215 1364 :
0215 1365 :
0215 1366 :
0215 1367 :
0215 1368 :
0215 1369 :
0215 1370 :
0215 1371 :
0215 1372 :
0215 1373 :
0215 1374 :
0219 1375 :
021C 1376 :
021F 1377 :
021F 1378 :
021F 1379 :
58 63 D0 021F 1380 : MOVL (R3),R8 : GET CONTENT OF PTE
02 19 0222 1381 : BLSS 10$ : CONTINUE IF VALID PAGE
52 52 D7 0224 1382 : DECL R2 : CLEAR VALID FLAG
50 52 E0 8F 8A 0226 1383 10$: BICB #^C<WSL$M_VALID!WSL$M_PAGTYP!WSL$M_PFNLOCK>,R2; ISOLATE INTERESTING
58 15 00 EF 022A 1384 : EXTZV #PTESV_PFN,#PTESV_PFN,R8,R0 : GET PFN FROM PTE
06 10 022F 1385 : BSBW OSDISPATCH : DISPATCH ON PAGE TYPE
0231 1386 :
E0 57 56 F3 0231 1387 : NOTVALID: AOBLEQ R6,R7,OWSLOOP : PROCESS ENTIRE WORKING SET LIST
21 11 0235 1388 : BRB PROCWRT : DONE WITH WORKING SET LIST, RESET HEADER
0237 1389 :
0237 1390 :
0237 1391 :
0237 1392 :
OWSLOOP: : OUTSWAP WS LOOP
: MOVL (R5)[R7],R2 : GET WORKING SET LIST ENTRY
: BLBC R2,NOTVALID : SKIP IF NOT VALID
: BSBW MMG$SVAPTECHK : CONVERT VA TO SVA OF PTE
R3 <- SVA OF PTE FOR VA IN R2
:
: MOVL (R3),R8 : GET CONTENT OF PTE
: BLSS 10$ : CONTINUE IF VALID PAGE
: DECL R2 : CLEAR VALID FLAG
: BICB #^C<WSL$M_VALID!WSL$M_PAGTYP!WSL$M_PFNLOCK>,R2; ISOLATE INTERESTING
: EXTZV #PTESV_PFN,#PTESV_PFN,R8,R0 : GET PFN FROM PTE
: BSBW OSDISPATCH : DISPATCH ON PAGE TYPE
:
: NOTVALID: AOBLEQ R6,R7,OWSLOOP : PROCESS ENTIRE WORKING SET LIST
: BRB PROCWRT : DONE WITH WORKING SET LIST, RESET HEADER
:
: OSDISPATCH:
: ASSUME WSL$V_VALID EQ 0
: ASSUME WSL$V_PAGTYP EQ 1
: ASSUME WSL$V_PFNLOCK EQ 4
```

```
0237 1393 ASSUME PFNSC_PROCESS EQ 0
0237 1394 ASSUME PFNSC_SYSTEM EQ 1
0237 1395 ASSUME PFNSC_GLOBAL EQ 2
0237 1396 ASSUME PFNSC_GBLWRT EQ 3
0237 1397 ASSUME PFNSC_PPGTBL EQ 4
0237 1398 ASSUME PFNSC_GPGTBL EQ 5
5D 6547 DE 0237 1399 MOVAL (R5)[R7],FP
0238 1400 CASE R2,<-
0238 1401 PROCTRANS,-
0238 1402 PROCVALID,-
0238 1403 WSLERR,-
0238 1404 WSLERR,-
0238 1405 GBLTRANS,-
0238 1406 GBLVALID,-
0238 1407 GBLWRTTRANS,-
0238 1408 GBLWRTVALID,-
0238 1409 PPGTBLTRANS,-
0238 1410 PPGTBLVALID,-
0238 1411 >,TYPE=B
05 0253 1412 RSB
0254 1413 SPACEFAIL:
0254 1414 BUG_CHECK INSSWPFIL,FATAL
0254 1415
0258 1416 PROCWRT:
0258 1417 MOVL PCBSL_WSSWP(R4),R2
0258 1418 BLEQ SPACEFAIL
025C 1419 MOVZWL PCBSW_APTCNT(R4),R0
0262 1420 MOVL R4,W^OSWPPCB
0267 1421 SUBL R11,R9
026A 1422 ROTL #<32-2>,R9,R4
026E 1423 CMPW R4,PHDSW_SWAPSIZE(R5)
0272 1424 BGTRU SPACEFAIL
0274 1425 MOVW R4,W^OSWPPGS
0279 1426 MOVL R11,R3
027C 1427 ADDL R0,R2
027C 1428
```

COMPUTE ADDRESS OF WSL ENTRY  
SWITCH ON WSL PAGE TYPE + PTE VALID BIT  
0 => PROCESS TRANSITION PAGE  
1 => PROCESS VALID PAGE  
2 => ??? BUGCHECK  
3 => ??? BUGCHECK  
4 => GLOBAL TRANSITION  
5 => GLOBAL VALID  
6 => GLOBAL WRITABLE TRANSITION  
7 => GLOBAL WRITABLE VALID  
8 => PROCESS PAGE TABLE TRANSITION  
9 => PROCESS PAGE TABLE VALID

SKIP PFN LOCK PAGES

INSUFFICIENT SWAP FILE SPACE

RESET PROCESS HEADER BASE REGISTERS  
GET SWAP ADDRESS  
BRANCH IF NO VBN AVAILABLE TO USE  
GET COUNT OF ACTIVE PAGE TABLES  
SAVE ADDRESS OF OUTSWAP PROCESS  
COMPUTE NUMBER OF PAGES \* 4  
DIVIDE COUNT BY 4  
DO WE HAVE ENOUGH SPACE FOR SWAP  
BRANCH IF NOT, THIS IS FATAL  
SAVE COUNT OF OUTSWAP PAGES  
SVAPTE FOR OUTSWAP I/O  
SKIP HEADER AND ACTIVE PAGE TABLES

```
027F 1431
027F 1432
027F 1433
027F 1434
027F 1435
027F 1436
027F 1437
027F 1438
0283 1439
0286 1440
0289 1441
028D 1442
028D 1443
028D 1444
0290 1445
0294 1446
0298 1447
029C 1448
02A4 1449
02A6 1450
02AC 1451
02B3 1452
02B8 1453
02BA 1454
02C1 1455
02C4 1456
02C9 1457
02CB 1458
02D5 1459
02D7 1460
02DA 1461
02DD 1462
02DF 1463
02E9 1464
02EB 1465
02EE 1466
02F0 1467
02F8 1468
02F8 1469
02FA 1470
02FC 1471
0302 1472
0305 1473
0308 1474
0308 1475
0314 1476

DO OUTSWAP I/O FOR PROCESS HEADER AND BODY

INCL W*SWPSGL_OSWPCNT ; ACCOUNT FOR OUTSWAP
BSBW SWPWRITE ; WRITE HEADER AND BODY
BLBS R0,20$ ; CONTINUE IF NO I/O ERROR
BUG_CHECK OUTSWPERR,FATAL ; **** OUT SWAP I/O ERROR

20$:
BSBW RELINIT ; INIT REGISTERS FOR RELEASE LOOP
MOVL PCBSL_PHD(R4),R5 ; GET POINTER TO PHD
MOVZWL PHDSW_PHVINDEX(R5),R8 ; GET PROCESS HEADER SLOT INDEX
BICL3 R10,(R9)+,R0 ; GET PAGE NUMBER TO RELEASE
CMPZV #PFNSV_PAGTYP,#PFNSS_PAGTYP,@W*PFNSAB_TYPE[R0],#PFNSC_GLOBAL ;
BEQL 80$ ; PAGE IS GLOBAL, COMPLEX CLEANUP
MOVL @W*PFNSAL_PTE[R0],R6 ; GET POINTER TO PAGE TABLE FOR PAGE
BICL #<PTESM_VALID!PTESM_MODIFY>,(R6); CLEAR VALID AND MODIFY
TSTW @W*PFNSAW_SWPVBN[R0] ; WAS I/O IN PROGRESS?
BEQL 40$ ; NO, DONT MARK PAGE MODIFIED
BISB #PFNSM_MODIFY,@W*PFNSAB_STATE[R0] ; MARK PAGE MODIFIED
CMPZV #PFNSV_LOC,#PFNSS_LOC,- ; IF THIS WAS READ IN PROGRESS
@W*PFNSAB_STATE[R0],#PFNSC_RDERR ; AND IS NOW PAGE READ ERROR
40$:
BNEQ 60$ ; AND IF THIS IS THE LAST REFERENCE
DECREf 60$
BNEQ 60$
MOVZBL #PFNSC_BADPAGLST,R2 ; THEN DIVERT THE PAGE TO
BSBW MMGSINSPFNT ; THE BAD PAGE LIST
BRB 60$
40$:
DECREf 55$ ; DECREMENT REFERENCE COUNT FOR PAGE
BNEQ 55$ ; NOT RELEASABLE YET
BSBW MMGSRELPFN ; RELEASE PFN AS APPROPRIATE
BRB 60$ ; GO FOR NEXT PAGE
50$:
CMPZV #PFNSV_LOC,#PFNSS_LOC,@W*PFNSAB_STATE[R0],- ;
#PFNSC_ACTIVE ; IS STATE ACTIVE?
BNEQ 60$ ; NO, THEN LEAVE UNCHANGED
INSV #PFNSC_RELPEND,- ; MAKE STATE BE RELEASE PENDING
#PFNSV_LOC,#PFNSS_LOC,@W*PFNSAB_STATE[R0] ; IF SOME I/O OUTSTANDING
60$:
SOBGTR R7,30$ ; NEXT PAGE IN LIST
BRW RELPHD ; RELEASE PROCESS HEADER IF POSSIBLE
80$:
DECSHR GTR=60$,- ; DECREASE SHARE COUNT FOR PAGE
BRB 40$ ; RELEASE PAGE TO FREE LIST IF REFCNT=0
```



## .SBTTL RELPHD - RELEASE PROCESS HEADER

++  
FUNCTIONAL DESCRIPTION:

RELPHD CHECKS THE REFERENCE COUNT ON THE PROCESS HEADER  
AND RELEASES THE PAGE TABLES FROM THE PROCESS HEADER WHEN ALL  
OF THEIR PAGES HAVE BEEN RELEASED. THE PAGE TABLES ARE FIRST WRITTEN  
TO THE SWAP IMAGE IF THEY ARE MARKED AS UPDATED.

## CALLING SEQUENCE:

BRW/JMP RELPHD

## INPUT PARAMETERS:

R8 - BALANCE SLOT INDEX FOR HEADER TO BE RELEASED

## OUTPUT PARAMETERS:

R0-R7,R9,R10 VOLATILE

## SIDE EFFECTS:

THE PAGE TABLES FROM THE PROCESS HEADER MAY BE WRITTEN TO THE  
SWAP IMAGE FOR THE PROCESS IF THEY HAVE BEEN UPDATED.

## RELPHD:

```
TSTW    @W*PHV$GL_REF(CBAS[R8])    : SEE IF PROCESS HEADER IS RELEASABLE
BEQL    5$                            : YES, FREE ACTIVE PAGE TABLES
BRW     OSWPEXIT                      : NO, TRY LATER
5$:     MOVL    W*SWP$GL_BSLOTSZ,R7    : SET ITERATION COUNT TO WHOLE BALANCE SLOT
        MULL3   R7,R8,R1              : GET LONG WORD OFFSET TO SLOT
        MOVAL   W*SWP$GL_BALSPT[R1],R6 : POINT TO BASE OF THIS SLOT
        OSINIT  : INIT REGISTERS FOR SCAN
        CVTWL   @W*PHV$GL_PIXBAS[R8],R4 : GET INDEX TO PROCESS IN SLOT
        BLSS    12$                   : BR IF DELETED PROCESS
        MOVL    @W*SCH$GL_PCBVEC[R4],R4 : AND TRANSLATE TO PCB ADDRESS
        MOVL    PCB$P_PHD(R4),R5       : GET PROCESS HEADER ADDRESS
        MOVL    R8,PCB$P_PHD(R4)       : INDICATE NO PHD FOR PROCESS
        SUBL    R5,PCB$P_POBR(R5)       : UNBIAS MEMORY MANAGEMENT BASE REGISTERS
        SUBL    R5,PCB$P_P1BR(R5)       : FOR BOTH P0 AND P1 SPACE
        BBCC    #PCB$P_PDRS,PCB$P_STS(R4),7$ : MARK PHD NON-RESIDENT
7$:     MOVL    PHD$P_WSLX(R5),AP       : GET POINTER TO WSLX SAVE AREA
        MOVAL   (R5)[AP],AP            : AND CONVERT TO BYTE ADDRESS
        MOVL    PHD$P_BAK(R5),FP        : GET POINTER TO BACKING STORE VECTOR
        MOVAL   (R5)[FP],FP            : AND CONVERT TO BYTE ADDRESS
        CLRW    PHD$P_EMPTYPG(R5)       : CLEAR COUNT OF EMPTY WSL PAGES
10$:    MOVL    (R6)+,(FP)+             : COPY ENTRY FROM SPT
        BLSS    15$                   : BR IF VALID
        BNEQ    11$                   : BR IF NOT EMPTY WSL PAGE
        INCW    PHD$P_EMPTYPG(R5)       : COUNT EMPTY WSL PAGES
11$:    CLRL    -4(R6)                 : ZAP INVALID ENTRY TO NO-ACCESS
        CLRW    (AP)+                 : AND CLEAR WSLX VALUE FOR PAGE
        BRB     20$                   :
20$:    BRW     DELPHD                 : FINISH DELETE FOR PROCESS
15$:    EXTZV   #PTESV_PFN,#PTES_PFN,-4(FP),R0 : GET PFN FOR VALID ENTRY
        BEQL    11$                   : DEMAND ZERO OR NULL PTE
        INSV    @W*PFNSAL_BAK[R0],#PTESV_PGFLVB,#PTES_PGFLVB,-4(FP) : SAVE BACKU
```

```
0000'DF48 B5 0316 1479
      03 13 0316 1480
      00E1 31 0316 1481
57 0000'CF D0 0316 1482
51 58 57 C5 0316 1483
56 0000'DF41 DE 0316 1484
      072F 30 0316 1485
54 0000'DF48 32 0316 1486
      43 19 0316 1487
54 0000'DF44 D0 0316 1488
      55 6C A4 D0 0316 1489
      6C A4 58 D0 0316 1490
      00C8 C5 55 C2 0316 1491
      00D0 C5 55 C2 0316 1492
      00 24 A4 12 E5 0316 1493
      5C 48 A5 D0 0316 1494
      5C 654C DE 0316 1495
      5D 44 A5 D0 0316 1496
      5D 654D DE 0316 1497
      00D8 C5 B4 0316 1498
      8D 86 D0 0316 1499
      10 19 0316 1500
      04 12 0316 1501
      00D8 C5 B6 0316 1502
      FC A6 D4 0316 1503
      8C B4 0316 1504
      23 11 0316 1505
      0084 31 0316 1506
50 FC AD 15 00 EF 0316 1507
      EE 13 0316 1508
FC AD 16 00 0000'DF40 F0 0316 1509
```

```
0391 1536
0391 1537
0391 1538
0391 1539
89 5A 50 C9 0397 1540
00 FC AD 1F E2 0398 1541
      C8 57 F5 03A0 1542 20$:
      59 58 C2 03A3 1543
      52 20 A4 D0 03A6 1544
0014 CF 54 D0 03AA 1545
54 59 1E 9C 03AF 1546
0012 CF 54 B0 03B3 1547
      53 58 D0 03B8 1548
      0000 CF D6 03BB 1549
      06CC 30 03BF 1550
      04 50 E8 03C2 1551
      068B 30 03C5 1552
      5B 6C A4 D0 03C9 1553 30$:
50 89 5A CB 03D0 1554 40$:
56 0000 DF40 D0 03D4 1555
      66 50 D0 03DA 1556
      0000 DF40 B7 03DD 1557
      04 13 03E2 1558
      03E4 1559
      53 56 D0 03E8 1560 50$:
      0683 30 03EB 1561
      66 D4 03EE 1562
      DD 57 F5 03F0 1563
0000 DF48 01 AE 03F3 1564
      0000 DF48 B4 03F9 1565
      6C A4 D4 03FE 1566
      0401 1567
      0401 1568
      05D3 31 0401 1570 OSWPEXIT:
      BRW SWAPRETRY

PFN REFERENCE -
MOVW <@W^PFNSAX WSLX[R0],(AP)+>,- ; AND WORKING SET LIST INDEX
LONG OPCODE=CVTLW,-
IMAGE=SYS NONPAGED
BISL3 R0,R10,(R9)+ ; SET INTO SWAPPER MAP
BBSS #PTESV_VALID,-4(FP),20$ ; MARK PAGE VALID FOR INSWAP PURPOSES
SOBGTR R7,10$ ; SCAN ENTIRE BALANCE SLOT
SUBL R11,R9 ; COMPUTE NUMBER OF PAGES * 4
MOVL PCB$L WSSWP(R4),R2 ; WORKING SET SWAP SLOT
MOVL R4,W^OSWPPCB ; SAVE PCB ADDRESS FOR SLOT OWNER
ROTL #<32-2>,R9,R4 ; DIVIDE COUNT BY 4
MOVW R4,W^OSWPPGS ; SAVE COUNT OF OUTSWAP PAGES
MOVL R11,R3 ; SET SVA OF MAP FOR I/O
INCL W^SWP$GL_HOSWPCNT ; ACCOUNT FOR HEADER OUTSWAP
BSBW SWPWRITE ; WRITE ACTIVE PAGE TABLES
BLBS R0,30$ ; CONTINUE IF NO ERROR
BUG CHECK APTWRTERR,FATAL ; **** ACTIVE PAGE TABLE SWAP I/O ERROR
BSBW RELINIT ; INIT REGISTERS FOR RELEASE LOOP
MOVL PCB$L PHD(R4),R8 ; RESTORE BALANCE SLOT INDEX
BICL3 R10,(R9)+,R0 ; ISOLATE PAGE FRAME NUMBER
MOVL @W^PFNSAL_PTE[R0],R6 ; GET PTE ADDRESS
MOVL R0,(R6) ; MAKE PTE CORRECT BUT INVALID
DECW @W^PFNSAW_REFCNT[R0] ; DROP REFERENCE COUNT
BEQL 50$ ; MUST BE ZERO
BUG CHECK APTREFHIGH,FATAL ; INCONSISTENT PAGE TABLE REFERENCE COUNT
MOVE R6,R3 ; SVAPTE FOR DELCON
BSBW RELDELPAGE ; RELEASE PAGE THROUGH DELCONPFN
CLRL (R6) ; SET NO ACCESS ON PFN
SOBGTR R7,40$ ; CONTINUE FOR ALL ACTIVE PAGE TABLES
MNEGW #1,@W^PHV$GL_REFCBAS[R8] ; MARK BALANCE SLOT AVAIL
CLRW @W^PHV$GL_PIXBAS[R8] ; AND SET PIX TO NULL
CLRL PCB$L_PHD(R4) ; AND SEVER CONNECTION WITH PROCESS

; OUTSWAP COMPLETE
; RETRY SWAP SCHEDULE AFTER OUTSWAP
```

```
0404 1573 .SBTTL DELPHD - DELETE PROCESS HEADER FOR DELETED PROCESS
0404 1574
0404 1575
0404 1576 FUNCTIONAL DESCRIPTION:
0404 1577 DELPHD IS ENTERED BY RELPHD IF THE PROCESS INDEX ASSOCIATED WITH
0404 1578 THE BALANCE SLOT IS NEGATIVE INDICATING THE PROCESS HAS BEEN DELETED.
0404 1579 NOW THAT THE REFERENCE COUNT FOR THE HEADER IS ZERO, ALL PAGES AND
0404 1580 BACKING STORE PAGES CAN BE RELEASED PERMITTING RELEASE OF THE BALANCE
0404 1581 SLOT. AT THIS POINT THE SPT ENTRIES ARE VALID WITH A PFN, DEMAND ZERO,
0404 1582 OR BACKING STORE ADDRESS FORM. THERE ARE NO REMAINING TRANSITION PAGES.
0404 1583
0404 1584 INPUT PARAMETERS:
0404 1585 R1 - PRODUCT OF SGNBC_BSLOTSZ * BALANCE SLOT INDEX
0404 1586 R6 - ADDRESS OF FIRST SPT ENTRY FOR THIS BALANCE SLOT
0404 1587 R7 - SGNBC_BSLOTSZ
0404 1588 R8 - BALANCE SLOT INDEX
0404 1589 R10 - MASK OF PTESM_VALID!PTESM_MODIFY!PTESC_ERKW
0404 1590
0404 1591
0404 1592 DELPHD:
0404 1593
0404 1594 ROTL #9,R1,R5 ; COMPUTE OFFSET TO PHD FROM BASE
0404 1595 ADDL W*SWP$GL BALBASE,R5 ; FORM PHD ADDRESS
0404 1596 MOVZBL PHD$B_PAGFIL(R5),R11 ; GET PAGING FILE NUMBER
0404 1597 10$: MOVL (R6)+,R0 ; GET PTE FROM SPT
0404 1598 BEQL 40$ ; BR IF EMPTY
0404 1599 BLSS 20$ ; BR IF VALID
0404 1600 BBS #PTESV_TYP1,R0,25$ ; BR IF TYPE 1 (BACKING STORE)
0404 1601 20$: BICL R10,R0 ; ISOLATE PFN
0404 1602 BEQL 30$ ; SKIP DEMAND ZERO PTE
0404 1603 MOVL @W*PFNSAL BAK[R0],R9 ; GET BACKUP ADDRESS
0404 1604 BICB #<<PTESM_VALID!PTESM_MODIFY>>-24>,-1(R6) ; CLEAR VALID AND MODIFY
0404 1605 BSBW REDELPAE ; RELEASE PAGE
0404 1606 MOVL R9,R0 ; GET BACKUP ADDRESS
0404 1607 25$: EXTZV #PTESV_PGFLVB,#PTESV_PGFLVB,R0,R0 ; GET PAG FIL VB
0404 1608 BEQL 30$ ; BR IF NONE
0404 1609 MOVL R11,R3 ; SET PAGING FILE NUMBER FOR RELEASE
0404 1610 BSBW MMG$DALCPAGFIL ; DEALLOCATE PAGING FILE PAGE
0404 1611 30$: CLRL -4(R6) ; ZAP SPT ENTRY
0404 1612 40$: SOBGTR R7,10$ ; RELEASE ENTIRE HEADER
0404 1613 INVALID ; INVALIDATE HEADER
0404 1614 MNEGW #1,@W*PHV$GL REFCBAS[R8] ; MARK SLOT EMPTY
0404 1615 CLRW @W*PHV$GL PIXBAS[R8] ; POINT OWNER PIX AT NULL PROCESS
0404 1616 DECB W*SCH$GW DELPHDCT ; ACCOUNT FOR DELETED HEADER
0404 1617 BRW SWAPRETRY ; AND RETRY SWAP ATTEMPT
```

55 51 09 9C 0404 1593  
55 0000'CF C0 0408 1594  
5B 1F A5 9A 040D 1595  
50 86 D0 0411 1596 10\$:  
2C 13 0414 1597  
04 19 0416 1598  
16 50 1A E0 0418 1599  
50 5A CA 041C 1600 20\$:  
1E 13 041F 1601  
59 0000'DF40 D0 0421 1602  
FF A6 84 8F 8A 0427 1603  
0642 30 042C 1604  
50 59 D0 042F 1605  
50 16 00 EF 0432 1606 25\$:  
06 13 0437 1607  
53 5B D0 0439 1608  
FBC1' 30 043C 1609  
FC A6 D4 043F 1610 30\$:  
CC 57 F5 0442 1611 40\$:  
0443 1612  
0000'DF48 01 AE 0448 1613  
0000'DF48 B4 044E 1614  
0000'CF B7 0453 1615  
057D 31 0457 1616



```
045A 1619 .SBTTL GBLTRANS/GBLVALID/GBLWRTVALID - HANDLE GLOBAL PAGES
045A 1620
045A 1621 :
045A 1622 :
045A 1623 :
045A 1624 :
045A 1625 :
045A 1626 :
045A 1627 :
045A 1628 GBLTRANS: ; TRANSITION GLOBAL PAGE
045A 1629 GBLWRTTRANS: ; TRANSITION WRITABLE GLOBAL PAGE
50 15 00 0000'DF40 F0 045A 1630 INSX @W*MMG$GL_GPTBASE[R0],#PTESV_PFN,#PTESV_PFN,R0 ; GET GLOBAL PFN FROM
0462 1631
0462 1632 .ENABL LSB
0462 1633 GBLDROP: ; DROP GLOBAL PAGE FROM WORKING SET
51 57 D0 0462 1634 MOVL R7,R1 ; GET WSL INDEX FOR RELEASE
53 DD 0465 1635 PUSHL R3 ; SAVE SVAPTE FOR FOLLOWING DECPTRF
FB96' 30 0467 1636 BSBW MMG$DELWSLEX ; DELETE WSL GIVEN INDEX
08 BA 046A 1637 POPR #*M<R3> ; RESTORE SVAPTE
FB91' 30 046C 1638 BSBW MMG$DECPTRF ; AND DROP PAGE TABLE REFERENCE
046F 1639 DECSHR GTR=10$,- ; DECREASE SHARE COUNT
046F 1640 IMAGE_FLAG=SYS_NONPAGED
047B 1641 PROCDROP:
047B 1642 DECFR GTR=20$ ; AND REF COUNT IF LAST SHARER
FB76' 30 0487 1643 BSBW MMG$RELPFN ; RELEASE PAGE IF LAST REFERENCE
048A 1644 10$:
048A 1645 RSB ; RETURN FOR NEXT PAGE
07 0000'DF40 03 00 05 048B 1646 20$: CMPZV #PFNSV_LOC,#PFNSS_LOC,@W*PFNSAB_STATE[R0] ;
0493 1647 #PFNSC_ACTIVE ; CHECK FOR ACTIVE STATE
08 12 0493 1648 BNEQ 30$ ; NO, THEN LEAVE STATE UNCHANGED
03 F0 0495 1649 INSX #PFNSC_RELPEND,- ; SET STATE TO RELEASE PENDING IF
0000'DF40 03 00 05 0497 1650 30$: RSB ; I/O OUTSTANDING
049D 1651 .DSABL LSB
049E 1652
049E 1653
049E 1654 :
049E 1655 :
049E 1656 :
049E 1657 GBLVALID: ; VALID GLOBAL PAGE
049E 1658 :
049E 1659 :
049E 1660 :
049E 1661 :
08 6D 05 E0 049E 1662 :
04A2 1663 :
04A2 1664 :
04A2 1665 :
04A2 1666 :
25 12 04A8 1667 BBS #WSLSV WSLOCK,(FP),10$ ; DON'T DROP PAGES LOCKED IN WORKING SET
04AA 1668 PFN_REFERENCE -
04AA 1669 CMPW <#1,@W*PFNSAx SHRCNT[R0]>,- ; IS THERE ACTIVE SHARING OF THIS PA
04AA 1670 LONG_OPCODE=CMPL,-
04AA 1671 IMAGE=SYS_NONPAGED
04AA 1672 BNEQ GBLWRTVALID ; YES, DROP IT AND REFAULT LATER
04AA 1673 :
04AA 1674 :
04AA 1675 :
89 5A 50 C9 04AA 1676 10$: BISL3 R0,R10,(R9)+ ; SET IN SWAPPER MAP FOR OUT SWAP
FB4F' 30 04AE 1677 BSBW MMG$DECPTRF ; DROP PAGE TABLE REFERENCE FOR PAGE
51 0000'DF40 0000'CF C3 04B1 1678 GBLRESET: ; RESET SLAVE PTE TO GPTX FORMAT
04B1 1679 SUBL3 W*MMG$GL_GPTBASE,@W*PFNSAL_PTE[R0],R1 ; GET GPTX FOR PAGE
```

```
51 51 1E 9C 04BA 1676 ROTL #<32-2>,R1,R1 ; AND CONVERT TO CORRECT SCALE
04BE 1677 ASSUME PTE$V_TYPO EQ PTE$S_GPTX ;
52 63 00 51 16 E2 04BE 1678 BBSS #PTE$V_TYPO,R1,10$ ; MARK AS GLOBAL
845FFFFFF 8F CB 04C2 1679 10$: BICL3 #<PTE$M_VALID ! - ; OBTAIN PERMANENT BITS FOR PTE
04CA 1680 PTE$M_TYPO ! - ; BY CLEARING ALL OTHERS
04CA 1681 PTE$M_TYPI ! - ;
63 52 51 C9 04CA 1682 PTE$M-PFN>,(R3),R2 ; TO FORM TRANSITION GLOBAL PTE
04CE 1683 BISL3 R1,R2,(R3) ; MUST SET ENTIRE PTE AT ONE TIME
05 04CE 1684 RSB ; SO THAT I/O CAN SEE CONSISTENT PTE
04CF 1685 ; RETURN FOR NEXT PAGE
04CF 1686 ;
04CF 1687 ;
04CF 1688 ;
04CF 1689 ;
04CF 1690 ;
04CF 1691 ;
04CF 1692 GBLWRTVALID: ; VALID WRITABLE GLOBAL PAGE
04CF 1693 BBCC #PTE$V_MODIFY,(R3),10$ ; TEST AND CLEAR MODIFY BIT IN SLAVE PTE
0000'DF40 07 63 1A E5 04CF 1694 BISB #PFNSM_MODIFY,@W^PFNSAB_STATE[RO] ; AND SAVE MODIFY STATE
80 8F 88 04D3 1694 BSBB GBLRESET ; RESET PTE
D5 10 04DA 1695 10$: BRB GBLDROP ; DELETE WORKING SET LIST ENTRY
84 11 04DC 1696
```

Address	Instruction	Comment
04DE 1699	.SBTTL	PROCTrans - PROCESS PAGE IN TRANSITION
04DE 1700		
04DE 1701		
04DE 1702		
04DE 1703		THIS PAGE IS IN TRANSITION DUE TO THE FACT THAT THE PAGE FAULT
04DE 1704		READ OPERATION HAS NOT YET COMPLETED. IT IS TREATED AS AN
04DE 1705		I/O IN PROGRESS PAGE.
04DE 1706		
04DE 1707	PROCTrans:	PROCESS PAGE IN TRANSITION
04DE 1708	CMPZV	#PFNSV LOC, #PFNSS LOC - IF THIS PAGE COULD NOT
04E1 1709		2W^PFNSAB_STATE[RO], #PFNSC_RDERR ; BE SUCCESSFULLY READ
04E6 1710	BNEQ	PROCVALID
04E8 1711	MOVL	R7, R1
04EB 1712	BSBW	MMGSDELWSLEX
04EE 1713	BRB	PROC DROP
04F0 1714		DROP IT FROM THE WORKING SET
04F0 1715		DELETE THE WSL ENTRY GIVEN WSL INDEX
04F0 1716		AND RELEASE THE PFN IF LAST REF
04F0 1717		
04F0 1718		PROCVALID HANDLES THE CASE OF A VALID PROCESS PAGE WHICH INCURS
04F0 1719		SOME SPECIAL PROCESSING IF THERE IS I/O IN PROGRESS. AN I/O IN
04F0 1720		PROGRESS PAGE IS SWAPPED WITH THE BODY OF THE PROCESS TO RESERVE
04F0 1721		SPACE FOR IT IN THE SWAP IMAGE AND IS LATER WRITTEN WITH CORRECT
04F0 1722		CONTENT BY THE MODIFIED PAGE WRITER TO THIS RESERVED SPACE IN THE
04F0 1723	PROCVALID:	PROCESS VALID PAGE
04F0 1724	.ENABL	LSB
04F0 1725	10\$:	
04F0 1726	BBSC	#PFNSV_MODIFY, 2W^PFNSAB_STATE[RO], 20\$ ; BR IF PAGE MODIFIED
04F7 1727	BBC	#PTESV_MODIFY, (R3), 30\$ ; BR IF PAGE NOT MODIFIED
04FB 1728	BBSS	#WSLSV_MODIFY, (FP), 30\$ ; SET WORKING SET MODIFIED BIT
04FF 1729	30\$:	
04FF 1730	CMPW	#1, 2W^PFNSAW_REFCNT[RO]
0505 1731	BEQL	40\$
0507 1732	BLBC	R2, SETWRTBAK
050A 1733	BBC	#WSLSV_MODIFY, (FP), 40\$
050E 1734	SETWRTBAK:	CHECK FOR I/O OUTSTANDING
050E 1735	SUBL3	R11, R9, R1
0512 1736	DIVL	#4, R1
0515 1737	MOVW	R1, 2W^PFNSAW_SWPVBN[RO]
051B 1738	40\$:	NO, NONE
051B 1739	BISL3	R0, R10, (R9)+
051F 1740		BRANCH IF TRANSITION PAGE
051F 1741		DONT WRITE UNMODIFIED PAGES
051F 1742		SET PAGE FOR WRITE BACK TO SWAP FILE
051F 1743		GET OFFSET TO PAGE IN SWAP MAP
051F 1744		SCALE BACK TO PAGE NUMBER
051F 1745		SET OFFSET INTO SWAP IMAGE LESS APTCNT
051F 1746		
051F 1747		PUT PAGE IN SWAPPER MAP
051F 1748		
051F 1749		SET DELETE CONTENT FLAG TO CAUSE PAGE TO BE PLACED AT HEAD
051F 1750		OF FREE PAGE LIST AND CONTENT FORGOTTEN.
051F 1751	DELCON:	BISB #PFNSM_DELCON, 2W^PFNSAB_STATE[RO] ; SET TO DELETE CONTENT
0525 1752	RSB	RETURN FOR NEXT PAGE
0526 1753	.DSABL	LSB
0526 1754		
0526 1755		
0526 1756	WSLERR:	BUG_CHECK IVWSETLIST, FATAL
0526 1757		INVALID WORKING SET LIST ENTRY



```
052A 1751 .SBTTL PAGE TABLE WORKING SET LIST ENTRIES
052A 1752
052A 1753 :
052A 1754 :
052A 1755 :
052A 1756 :
052A 1757 :
052A 1758 :
052A 1759 PPGTBLTRANS: : TRANSITION PAGE TABLE
052A 1760 PPGTBLVALID: : VALID PAGE TABLE
052A 1761 INCW PCBSW APTCNT(R4) : ACCUMULATE ACTIVE PAGE TABLE COUNT
052D 1762 SUBL R5,(FP) : UNBIAS WSL VA FOR PAGE TABLE
0530 1763 BBSS #VASV_SYSTEM,(FP),10$ : BUT FORCE SYSTEM BIT ON IN VA
0534 1764 10$: RSB : RETURN
```

30 A4 B6  
6D 55 C2  
00 6D 1F E2  
05

```
0535 1767 .SBTTL INSWAP
0535 1768
0535 1769
0535 1770 -----
0535 1771 PERFORM REQUESTED INSWAP
0535 1772
0535 1773 INPUTS:
0535 1774 R4 - PCB ADDRESS OF INSWAP CANDIDATE
0535 1775 -----
0535 1776 INSWAP:
0535 1777 MOVL PCB$$_PHD(R4),R5 ; PERFORM INSWAP
0535 1778 BEQL 10$ ; GET CURRENT PROCESS HEADER SLOT
0535 1779 MOVZWL PHD$$_PHVINDEX(R5),R8 ; NONE, MUST ALLOCATE ONE
0535 1780 BRB 40$ ; GET BALANCE SLOT INDEX
0535 1781 10$: CLRL R8 ; AND CONTINUE
0535 1782 20$: TSTW @W$$_PHV$$_GL_REF(BAS[R8]) ; INIT INDEX FOR BALANCE SLOT SEARCH
0535 1783 BLSS 30$ ; CHECK FOR EMPTY
0535 1784 AOBLS SGN$$_GL_BALSETCT,R8,20$ ; YES, GOT ONE
0535 1785 ROTL #31,#1,FP ; TRY ALL BALANCE SET SLOTS
0535 1786 ; SET FLAG TO PERMIT OUTSWAPPING
0535 1787 ; OF PROCESSES
0535 1788 30$: BRW OUTSWAP ; OUTSWAP IF NECESSARY TO GET SLOT
0535 1789 MOVW PCB$$_PID(R4),@W$$_PHV$$_GL_PXBAS[R8] ; SET PIX FOR BALANCE SET SL
0535 1790 CLRW @W$$_PHV$$_GL_REF(BAS[R8]) ; AND BUMP REFERENCE COUNT
0535 1791 MULL3 W$$_SWP$$_GL_BSLOTSZ,R8,R0 ; COMPUTE BALANCE SLOT OFFSET
0535 1792 ROTL #9,R0,PCB$$_PHD(R4) ; MAKE BYTE OFFSET
0535 1793 40$: CLRL R9 ; POSITIVE UNTIL I/O COMPLETE
0535 1794 50$: BSBW MMG$$_ALLOC PFN ; INITIALIZE SWAPPER MAP INDEX
0535 1795 TSTL R0 ; ALLOCATE A PAGE
0535 1796 BGEQ 60$ ; MAKE SURE IT WAS ALLOCATED
0535 1797 BUG CHECK INSN$$_FREPAG,FATAL ; YES, CONTINUE
0535 1798 INCQ @W$$_PFNS$$_AW_REFCT[R0] ; INSUFFICIENT FREE PAGES
0535 1799 MOVW #PFNS$$_C_ACTIVE,@W$$_PFNS$$_AB_STATE[R0] ; REFERENCE PAGE
0535 1800 BISL3 #<PTES$$_C_ERKW!PTESM_VALID>,R0,@W$$_SWP$$_GL_MAP[R9] ; AND MARK IT ACTIVE
0535 1801 AOBLS R10,R9,50$ ; MARK VALID, WRITABL
0535 1802 CLRL @W$$_SWP$$_GL_MAP[R9] ; REPEAT FOR ACL REQUIRED PAGES
0535 1803 ; PUT STOPPER IN LIST
0535 1804 ;
0535 1805 ; ALL PAGES HAVE NOW BEEN ACQUIRED AND A BALANCE SET SLOT
0535 1806 ; ALLOCATED. THE INSWAP I/O OPERATION CAN NOW BE PERFORMED.
0535 1807 INCW W$$_SWP$$_GL_BALCNT ; ADD ONE PROCESS TO BALANCE SET
0535 1808 MOVL R4,W$$_SWP$$_GL_INPCB ; SAVE POINTER TO IN SWAP PCB
0535 1809 MOVL R10,W$$_SWP$$_GL_ISPAGCNT ; SAVE COUNT OF ALLOCATED PAGES
0535 1810 MOVW R8,W$$_SWP$$_GL_IBALSETX ; AND BALANCE SET SLOT NUMBER
0535 1811 -----
0535 1812 PERFORM INSWAP I/O OPERATION
0535 1813 -----
0535 1814
0535 1815
0535 1816
0535 1817 MOVL PCB$$_WSSWP(R4),R2 ; GET SWAP IMAGE DISK ADDRESS
0535 1818 BLEQ COPYSHELL ; BRANCH IF SHELL IN SWAP
0535 1819 BBC #PCB$$_V_PHDRS,PCB$$_STS(R4),70$ ; SWAP EVERYTHING IF HEADER NON-RES
0535 1820 MOVZWL PCB$$_W_XPTCNT(R4),R0 ; GET ACTIVE PAGE TABLE COUNT
0535 1821 ADDL R0,R2 ; ADD PAGE TABLE COUNT
0535 1822 MOVAL @W$$_SWP$$_GL_MAP,R3 ; SVA OF PAGE TABLE FOR I/O
0535 1823 MOVL R10,R4 ; NUMBER OF PAGES TO READ
```

```
0000'CF 5A C0 05C9 1824 ADDL2 R10,W^SWP$GL_ISWPPAGES : UPDATE TOTAL PAGES INSWAPPED
0000'CF D6 05CE 1825 INCL W^SWP$GL_ISWPCNT : BUMP INSWAP COUNTER
04B3 30 05D2 1826 BSBW SWPREAD : PERFORM READ
04 50 E8 05D5 1827 BLBS RO,808 : BRANCH IF NO ERROR IN READ
00B9 31 05D8 1828 BUG_CHECK INSWAPERR,FATAL : **** BUGCHECK ON I/O ERROR
05DC 1829 BRW- SETUP : SET UP PROCESS IN BALANCE SLOT
05DF 1830
05DF 1831
54 0000'CF DE 05DF 1832 COPYSHELL:
55 6C A4 D0 05E4 1833 MOVAL W^MMG$AL_SYSPCB,R4 : ADDRESS OF SYSTEM PCB
56 0000'CF D0 05E8 1834 MOVL PCB$P_PHD(R4),R5 : ADDRESS OF SYSTEM PROCESS HEADER
52 00000000'EF 9E 05ED 1835 MOVL W^SWP$GL_SHELLIO,R6 : GET I/O PAGE COUNT FOR SHELL
FA09' 30 05F4 1836 MOVAB L^SWP$GL_SHELLBAS,R2 : GET ADDRESS OF SHELL
0048 8F BB 05F7 1837 BSBW MMG$SVAPTECHK : GET ADDRESS OF PAGE TABLE ENTRY
58 D4 05FB 1838 PUSHF #M<R3,R6> : SAVE SVAPTE AND PAGE COUNT FOR LATER
57 0103 8F 3C 05FD 1839 CLRL R8 : SET FLAG INDICATING NO I/O NEEDED
0602 1840 MOVZWL #256+3,R7 : SET FLAGS TO LOCK ONLY VALID OR
0602 1841 : TRANSITION PAGES AND CREATE OTHERS
52 02 90 0602 1842 : WITHOUT ZEROING THE PHYSICAL PAGE
0000'CF 00' 88 0605 1843 MOVAB #WSL$C_SYSTEM,R2 : SET PAGE TYPE IN LOW BITS
F9F3' 30 060A 1844 BISB S^MMG$M_NOWAIT,W^MMG$GB : FREWFLGS ; PREVENT FREWLSE MWAIT
04 50 E8 060D 1845 BSBW MMG$IOLOCKPAG : LOCK THE PAGE INTO SYSTEM WORKING SET
0610 1846 BLBS RO,208 : BRANCH IF SUCCEEDED
58 50 88 0614 1847 BUG_CHECK INSNFREPAGE,FATAL : INSUFFICIENT FREE PAGES
52 0200 C2 9E 0617 1848 BISB RO,R8 : SET FLAG (BIT 1) IF WE HAVE TO I/O IT
53 04 C0 061C 1849 MOVAB 512(R2),R2 : BUMP VA TO NEXT PAGE
E8 56 F5 061F 1850 ADDL #4,R3 : BUMP PTE TO NEXT ENTRY
18 BA 0622 1851 SOBGTR R6,108 : LOOP THROUGH THE PAGES
16 58 01 E1 0624 1852 POPR #M<R3,R4> : RECOVER SVAPTE AND PAGE COUNT
0000'CF 00' 8A 0628 1853 BBC #1,R8,408 : BRANCH IF ALL SHELL PAGES IN MEMORY
52 D4 062D 1854 BICB S^MMG$M_NOWAIT,W^MMG$GB : FREWFLGS ; ALLOW FREWLSE MWAIT
0456 30 062F 1855 CLRL R2 : SHELL IS PAGE FILE 0 AND VBN 0
04 50 E8 0632 1856 BSBW SWPREAD : PERFORM SHELL READ
0635 1857 BLBS RO,308 : BRANCH IF NO ERROR IN READ
0000'CF 00' 88 0639 1858 BUG_CHECK INSWAPERR,FATAL : **** BUGCHECK ON I/O ERROR
063E 1859 BISB S^MMG$M_NOWAIT,W^MMG$GB : FREWFLGS ; PREVENT FREWLSE MWAIT
56 0000'CF D0 0641 1860 SETIPL #IPL$_ASTDEL : ALLOW RESCHEDULE AND PAGEFAULTS WHILE
57 00000000'EF 9E 0646 1861 MOVAB W^SWP$GL_SHELLIO,R6 : GET I/O PAGE COUNT FOR SHELL
50 56 09 78 064D 1862 MOVAB L^SWP$GL_SHELLBAS,R7 : GET ADDRESS OF SHELL
51 D4 0651 1863 ASHL #9,R6,R0 : GET BYTE COUNT
61 67 50 28 0653 1864 CLRL R1 : FORM DESTINATION VA
0657 1865 MOVCL RO,(R7),(R1) : COPY THE SHELL TO LOCATION 0
0000'CF 00' 8A 065A 1866 SETIPL #IPL$_SYNCH : BACK TO BLOCKING IPL
51 56 7D 065F 1867 BICB S^MMG$M_NOWAIT,W^MMG$GB : FREWFLGS ; ALLOW FREWLSE MWAIT
F998' 30 0662 1868 MOVQ R6,R1 : SET UP COUNT AND VA OF SHELL AGAIN
F998' 30 0665 1869 BSBW MMG$SVAPTECHK : GET ADDRESS OF PAGE TABLE ENTRY
0668 1870 BSBW MMG$UNLOCK : DROP THE REFERENCE COUNTS
0668 1871 : CONTINUE PROCESS CREATION
```



```
0668 1874
0668 1875
0668 1876
0668 1877
0668 1878
0668 1879
0668 1880
0668 1881
0668 1882
0668 1883
0670 1884
0673 1885
0675 1886
0678 1887
067E 1888
0680 1889
0683 1890
0689 1891
0689 1892
0689 1893
068E 1894
0694 1895
069A 1896
069D 1897
06A2 1898
06A6 1899
06A9 1900
06A9 1901
06A9 1902
06A9 1903
06A9 1904
06AD 1905
06B3 1906
06B9 1907
06BE 1908
06C3 1909
06C7 1910
06CC 1911
06CC 1912
06CC 1913
06CC 1914
06D0 1915
06D3 1916
06D3 1917
06D3 1918
06D3 1919
06D3 1920
06D3 1921
06D3 1922
06D3 1923
06D3 1924
06DA 1925
06DD 1926
06E2 1927
06E9 1928
06EC 1929
06EE 1930

54      03F6      30
      0000'CF      D0
      6C A4      D5
      06      19
6C A4    0000'CF      C0
      20 A4      D5
      09      14
      00000000'9F      16

57      58      0000'CF      3C
      58      0000'CF      C5
57      0000'DF47      DE
      53      57      D0
      1C 24 A4      12      E2
      0000'CF      D6
      0346      30

      42 A5      58      B0
00C8 C5      6C A4      C0
00D0 C5      6C A4      C0
      00 36 A5      03      E2
50      67      15      00      EF
      50      50      09      9C
      18 A4      78 A0      9E

      55      6C A4      D0

52      00000000'EF      D0
      F923'      30
      52      0000'CF      D0
51      F0000000 8F      D0
      50      87      D0
      04      19      06EC
      83      D4      06EE

SET UP PROCESS IN BALANCE SET SLOT

SETUP:
BSBW OSINIT ; SETUP INSWAP PROCESS
MOVL W*SWP$GL_INPCB,R4 ; INIT REGISTERS
TSTL PCB$$_PHD(R4) ; GET PCB ADDRESS OF INSWAP PROCESS
BLSS 10$ ; CHECK FOR NEWLY ALLOCATED PHD
ADDL W*SWP$GL_BALBASE,PCB$$_PHD(R4) ; AND SET ADDRESS IN PCB
TSTL PCB$$_W$SWP(R4) ; CHECK FOR SHELL INSWAP
BGTR NOTSHELL ; BR IF NOT
INVALID ; CLEAR TRANSLATION BUFFER
JSB @#SWP$SHELINIT ; CALL SHELL INITIALIZATION
; WHICH RETURNS WITH A FULLY INITED PHD

NOTSHELL:
MOVZWL W*SWP$GW_IBALSETX,R8 ; AND BALANCE SET INDEX
MULL3 W*SWP$GL_B$LOTSZ,R8,R7 ; COMPUTE OFFSET TO THIS SLOT
MOVAL @W*SWP$GL_B$ALSPT(R7),R7 ; FORM BASE ADDRESS OF MAP FOR SLOT
MOVL R7,R3 ; NOW POINT TO PROCESS HEADER
BBSS #PCB$V_PHDR$S,PCB$$_STS(R4),5$ ; SKIP IF PROCESS HEADER STILL RESID
INCL W*SWP$GL_HISWPCNT ; COUNT SWAPS INCLUDING HEADER
BSBW FILLPHD ; SET INTO SPT ENTRIES

FILLPHD RETURNS WITH R5 POINTING TO THE PROCESS HEADER POSITION
WITHIN ITS PO SPACE.

MOVW R8,PHD$W_PHVINDE(R5) ; SET BALANCE SLOT INDEX
ADDL PCB$$_PHD(R4),PHD$$_POBR(R5) ; RELOCATE PO BASE REGISTER
ADDL PCB$$_PHD(R4),PHD$$_P1BR(R5) ; RELOCATE P1 BASE REGISTER
BBSS #PHD$V_NOACCVIO,PHD$W_FLAGS(R5),5$ ; INDICATE PHD INSWAP TO PAGER
EXTZV #0,#PTES$_PFN,(R7),R0 ; GET PHYSICAL ADDRESS OF PCB
ROTL #9,R0,R0 ; AND SET IN SOFTWARE PCB
MOVAB PHD$$_PCB(R0),PCB$$_PHPCB(R4) ; ADD OFFSET TO HW PCB

NOW SET PAGES FROM WORKING SET LIST INTO PAGE TABLE ENTRIES

MOVL PCB$$_PHD(R4),R5 ; GET PROCESS HEADER ADDRESS
INVALID ; CLEAR TRANSLATION BUFFER TO SEE IT

A WINDOW IN P1 SPACE IS DOUBLE MAPPED TO ALL OF THE PROCESS
HEADER EXCEPT FOR THE PAGE TABLES. THIS PERMITS REFERENCE TO
MOST OF THE PROCESS HEADER WHILE RUNNING AT IPL LESS THAN THE
SCHEDULER. TO REFERENCE THE PROCESS HEADER IN SYSTEM SPACE
A PROCESS(OTHER THAN THE SWAPPER) MUST RAISE TO IPL$_SYNCH.

MOVL SWP$GL_PHDBASVA,R2 ; VIRTUAL ADDRESS OF PHD WINDOW
BSBW MMG$SVAPTECHK ; GET POINTER TO WINDOW PTE
MOVL W*SGN$GL_PHPAGCT,R2 ; SET COUNT OF PAGES FOR WINDOW
MOVL #<PTESC_ORKW!PTESM_VALID>,R1 ; SKELETON PTE
10$: MOVL (R7)+,R0 ; GET SWAPPER PTE FOR PHD
BLSS 20$ ; BR IF VALID PAGE
CLRL (R3)+ ; NO, SET NO ACCESS
```

51	15	00	08	11	06F0	1931	
		83	50	F0	06F2	1932	208:
		EC	51	D0	06F7	1933	
			52	F5	06FA	1934	308:
					06FD	1935	
					06FD	1936	
					06FD	1937	
					06FD	1938	
					06FD	1939	
					06FD	1940	
					06FD	1941	
					06FD	1942	
					06FD	1943	
5B	01	1F	9C	06FD	1944		
56	08	A5	3C	0701	1945		
57	12	A5	3C	0705	1946		
				0709	1947		
	52	6546	D0	0709	1948	WSLO	
		12	10	070D	1949		
F6	56	57	F3	070F	1950		
50	89	5A	CB	0713	1951	58:	
		05	13	0717	1952		
		0358	30	0719	1953		
		F5	11	071C	1954		
		0261	31	071E	1955	78:	
				0721	1956		
	0A	52	E6	0721	1957	108:	
			05	0724	1958	158:	
	52	55	C0	0725	1959	178:	
6546	52	5B	C9	0728	1960		
			05	072D	1961	188:	
				072E	1962	208:	
		F5	19	072E	1963		
		F8CD	30	0730	1964		
				0733	1965		
				0733	1966		
				0733	1967		
				0733	1968		
				0733	1969		
				0733	1970		
				0733	1971		
				0733	1972		
				0733	1973		
				0733	1974		
				0733	1975		
				0733	1976		
				0733	1977		
50	89	5A	CB	0733	1978		
		04	12	0737	1979		
				0739	1980		
	5C	63	D0	073D	1981	308:	
		0B	18	0740	1982		
				0742	1983		
				0742	1984		IF
				0742	1985		OU
				0742	1986		BE
				0742	1987		AC

```

BRB      30$
INSV     R0,#PTESV_PFN,#PTESV_PFN,R1      : AND INSERT PFN INTO WINDOW PTE
MOVL     R1,(R3)+                          : STORE IN WINDOW AND ADVANCE TO NEX PTE
SOBGR    R2,10$                            : MAP ENTIRE PHD WINDOW

THE REMAINING LIST OF PAGES READ BY THE SWAPPER ARE NOW PROCESSED
ACCORDING TO THE CONTENT OF THE WORKING SET LIST IN THE HEADER OF
THE INSWAP PROCESS.  THE DISPOSITION OF EACH INSWAP PAGE DEPENDS
ON ITS TYPE AND WHETHER THE PAGE IS ALREADY PRESENT IN WHICH CASE
THE NEW, REDUNDANT COPY IS DISCARDED.  SHARED PAGES READ FROM THE
SWAP IMAGE WHICH ARE NOT ALREADY RESIDENT BECOME THE MASTER COPY
AS WELL AS SATISFYING THE REQUIREMENT OF THE INSWAP PROCESS.

ROTL     #PTESV_VALID,#1,R11              : FORM VALID MASK
MOVZWL   PHD$W_WSLIST(R5),R6              : INDEX TO START OF PERM ENTRIES
MOVZWL   PHD$W_WSLAST(R5),R7              : POINTER TO LAST WS ENTRY
ENABL     LSB                             :
MOVL     (R5)[R6],R2                      : GET A WORKING SET ENTRY
BSBB     10$                             : AND PROCESS IT
AOBLEQ    R7,R6,WSLOOP                    : SCAN ENTIRE WORKING SET LIST
BICL3    R10,(R9)+,R0                     : GET AND RELEASE EXCESS PAGES
BEQL     7$                              : BR IF NO MORE
BSBW     RELPAGE                          : RELEASE AN EXCESS PAGE
BRB      5$                              : AND TRY FOR ANOTHER
BRW      SETASTLVL                        : END OF WORKING SET LIST
ASSUME    WSL$V_VALID EQ 0                : FOR USE OF BLBS
BLBS     R2,20$                          : CHECK FOR VALIDITY, BR IF VALID
RSB      R5,R2                           : GET NEXT WSL ENTRY IF NOT VALID
ADDL     R5,R2                           : REBIAS VA FOR WSL ENTRY
BISL3    R11,R2,(R5)[R6]                 : AND SET SYSTEM BIT IN VA
RSB      R5,R2                           : NEXT WORKING SET LIST ENTRY

BLSS     17$                              : SKIP PAGE TABLE ENTRIES
BSBW     MMG$SVAPTECHK                    : GET SVA OF PTE FOR PAGE

R0 - ALL BITS EXCEPT PFN FIELD ARE CLEAR
R2 - WS LIST ENTRY
R3 - SVA OF PTE
R4 - INSWAP PROCESS PCB
R5 - PHD ADDRESS FOR INSWAP PROCESS
R6 - WORKING SET INDEX
R7 - END INDEX TO WORKING SET
R8 - BALANCE SET SLOT INDEX
R9 - ADDRESS OF PHYSICAL PAGE POINTER IN SWP$AL_MAP
R10 - PTES$ ERKW!PTES$M_VALID!PTES$M_MODIFY
R11 - CONSTANT PFN$M_VALID

BICL3    R10,(R9)+,R0                    : GET PFN FROM MAP
BNEQ     30$                             : GOT A GOOD PFN
BUG CHECK ZEROPAGE,FATAL                 : ZERO PAGE TABLE ENTRY FROM SWAP MAP
MOVL     (R3),AP                          : GET CONTENT OF PTE
BGEQ     35$                             : PTE VALID => PFN LOCK, NOT SWAPPED

E PAGE IS VALID, IT MUST HAVE BEEN LOCKED IN MEMORY AND WAS IGNORED AT
AP.  THE REDUNDANT PAGE ALLOCATED FOR THIS WORKING SET LIST ENTRY MUST
BE RELEASED AFTER ALL WORKING SET LIST ENTRIES ARE PROCESSED.  NO OTHER
ACTION IS NEEDED SINCE THE PTE FOR THE LOCKED PAGE IS ALREADY CORRECT.

```

SWA  
Sym[illegible]



				0742	1988	:	THE PFN THAT WOULD HAVE MATCHED THIS ENTRY REALLY BELONGS TO THE NEXT
				0742	1989	:	WSL WITHOUT PFNLOCK SET, SO THE MAP POINTER IN R9 MUST BE BACKED UP FOR
				0742	1990	:	ANOTHER TRY.
				0742	1991	:	
03	52	04	E1	0742	1992	BBC	#WLSV_PFNLOCK,R2,32\$ : ERROR IF PAGE NOT LOCKED IN MEMORY
		79	D5	0746	1993	TSTL	-(R9) : BACK UP IN SWAPPER MAP
			O5	0748	1994	RSB	: AND CONTINUE WITH NEXT WSL
				0749	1995	BUG_CHECK	ICPAGELOC,FATAL : INCONSISTENT PTE/WSL
71	5C	1A	E1	074D	1996	BBC	#PTESV_TYPI,AP,NTYP1 : BR IF NOT TYPE 1
0000'DF40		53	D0	0751	1997	MOVL	R3,@W^PFNSAL_PTE[R0] : NOTE LOCATION OF PTE
				0757	1998		PFN_REFERENCE -
				0757	1999	MOVW	<R6,@W^PFNSAx_WSLX[R0]>,- ; AND POSITION IN WORKING SET
				0757	2000		LONG_OPCODE=MOVZWL,-
				0757	2001		IMAGE=SYS_NONPAGED
51	63	17	00	EF	075D	EXTZV	#PFNSV_BAK,#PFNSS_BAK,(R3),R1 : GET BACKING ADR FROM PTE
		06	51	EO	0762	BBS	#PTESV_TYPO,R1,40\$ : BR IF SECTION ADDRESS
51	08	18	1F	A5	FO	INSV	PHDSB_PGFLX(R5),#PFNSV_PGFLX,#PFNSS_PGFLX,R1
					076C		: SET PAGINING FILE NUMBER
0000'DF40		51	D0	076C	2006	40\$: MOVL	R1,@W^PFNSAL_BAK[R0] : STORE BACKING ADDRESS
0000'DF40		07	90	0772	2007	MOVW	#PFNSC_ACTIVE,@W^PFNSAB_STATE[R0]; SET PAGE ACTIVE
				0778	2008	RECONNECT:	: RECONNECT TO PAGE
51	63	867FFFFFFF	8F	CB	0778	BICL3	#<PTESM_PROT!PTESM_OWN>,(R3),R1 : RETAIN PERMANENT BITS
		52	6546	DE	0780	MOVAL	(R5)[R6],R2 : GET ADDRESS OF WORKING SET LIST ENTRY
		07	62	08	E5	BBCC	#WLSV_MODIFY,(R2),50\$ : CHECK FOR MODIFIED AND CLEAR
0000'DF40		80	8F	88	0788	BISB	#PFNSM_MODIFY,@W^PFNSAB_STATE[R0] : RECORD MODIFY STATE
					078F	2013	50\$: BISL
					0792	2014	R11,R1 : SET VALID BIT FOR PTE
					0796	2015	RO,R1,(R3) : MERGE BITS WITH PFN AND STORE IN PGTL
50	15	00	0000'DF41	FO	0798	2016	EXTZV #VASV_VPN,#VASS_VPN,R3,R1 : GET VPN OF PAGE TABLE
					07A3	2017	INSV @W^MMG\$GL_SPTBASE[R1],#0,#PTESS_PFN,RO : GET PT PFN
					07A3	2018	: ASSUMES HIGH ORDER BITS OF RO ARE CLEAR
					07A3	2019	PFN_REFERENCE -
					07A3	2020	<@W^PFNSAx_SHRCNT[R0]>,- ; CHECK FOR FIRST ACTIVE PAGE
					07A3	2021	LONG_OPCODE=TSTL,-
					07A8	2022	IMAGE=SYS_NONPAGED
					07AA	2023	60\$ : NO, JUST RAISE SHARE COUNT FOR PT
					07AA	2024	PFN_REFERENCE -
					07AA	2025	<@W^PFNSAx_WSLX[R0],R1>,- ; GET INDEX TO WSL ENTRY FOR PAGE TA
					07AA	2026	LONG_OPCODE=MOVL,-
					07B0	2027	IMAGE=SYS_NONPAGED
6541		20	C8	07B0	2027	BISL	#WLSM_WLOCK,(R5)[R1] : AND MARK IT LOCKED IN WORKING SET
		70	A5	B6	07B4	2028	INCW PHDSW_PTCONTACT(R5) : COUNT ANOTHER ACTIVE PAGE TABLE
0000'DF48			B6	07B7	2029	INCW	@W^PHVSGL_REFCBAS[R8] : RAISE REFERENCE COUNT OF BALANCE SLOT
					07BC	2030	60\$: :
					07BC	2031	PFN_REFERENCE -
					07BC	2032	<@W^PFNSAx_SHRCNT[R0]>,- ; INDICATE ANOTHER ACTIVE PAGE FOR P
					07BC	2033	LONG_OPCODE=INCL,-
					07BC	2034	IMAGE=SYS_NONPAGED
					07C1	2035	RSB : RETURN TO GET NEXT WSL ENTRY
					07C2	2036	.DSABL LSB :
					07C2	2037	GLOBAL OR TRANSITION
					07C2	2038	NTP1: BBS #PTESV_TYPO,AP,11\$ : BR IF GLOBAL PAGE
5D	5C	2D	5C	16	EO	07C6	EXTZV #PTESV_PFN,#PTESS_PFN,AP,FP : GET OLD PFN IF ANY
			00	EF	07C6	2039	BEQL 12\$ : BR IF ZERO PAGE (BUG CHECK)
			29	13	07CD	2040	:
					07CD	2041	:
					07CD	2042	:
					07CD	2043	RELEASE PFN FOR PAGE ALREADY PRESENT
					07CD	2044	BSBW RELPAGE : RELEASE DUPLICATE PAGE

[illegible]



		50	50	D0	07D0	2045
52	0000'DF40	03	00	EF	07D3	2046
					07DB	2047
					07DB	2048
					07DB	2049
					07DB	2050
					07DB	2051
					07DB	2052
					07DB	2053
					07DB	2054
					07DB	2055
					07DB	2056
					07DB	2057
					07DB	2058
					07DB	2059
					07DB	2060
					07DB	2061
					07DB	2062
					07DB	2063
					07EF	2064
					07EF	2065
					07F3	2066
		0072		31	07F3	2067
					07F6	2068
					07FA	2069
		53		DD	07FA	2070
		F801'		30	07FC	2071
		08		BA	07FF	2072
		52	6546	DE	0801	2073
		63	5B	C8	0805	2074
					0808	2075
	0000'DF40		07	88	0808	2076
				8A	080E	2077
					080F	2078
	0000'DF40	90	8F		080F	2079
		0000'DF40		B6	0815	2080
		0000'DF40		B4	081A	2081
		52	6546	DE	081F	2082
		07	62	E5	0823	2083
	0000'DF40		80	88	0827	2084
					082E	2085
					082E	2086
					082E	2087
					082E	2088
				05	0834	2089
					0835	2090
					0835	2091
					0835	2092
					0835	2093
					0835	2094
					0835	2095
					0835	2096
	B3 0000'DF40		05	E0	0835	2097
			53	DD	083C	2098
			F7BF'	30	083E	2099
			08	BA	0841	2100
		52	6546	DE	0843	2101

```

MOVL    FP,R0                ; GET SAVED PFN
EXTZV   #PFNSV LOC,#PFNSS LOC,@W^PFNSAB_STATE[R0],R2
ASSUME   PFNSC_FREPAGLST EQ 0
ASSUME   PFNSC_MFYPAGLST EQ 1
ASSUME   PFNSC_BADPAGLST EQ 2
ASSUME   PFNSC_RELPEND   EQ 3
ASSUME   PFNSC_RDERR     EQ 4
ASSUME   PFNSC_WRTINPROG EQ 5
ASSUME   PFNSC_RDINPROG  EQ 6
ASSUME   PFNSC_ACTIVE    EQ 7
CASE     R2,<-                ; DISPATCH ON PAGE LOCATION
20$,-    0=> FREE PAGE LIST
20$,-    1=> MODIFIED PAGE LIST
60$,-    2=> BAD PAGE LIST, PAGE READ/WRITE ERR
30$,-    3=> RELEASE PENDING
10$,-    4=> PAGE READ ERROR
30$,-    5=> WRITE IN PROGRESS
40$,-    6=> READ IN PROGRESS
30$>     7=> ACTIVE ( I/O NOT YET COMPLETE

BUG_CHECK ICPAGELOC,FATAL    ; INCONSISTENT PAGE LOCATION

BRW      GLOBAL              ; GLOBAL PAGE
BUG_CHECK ZEROPAGE,FATAL    ; ZERO PFN IN PTE

PUSHL    R3                  ; SAVE SVAPTE
BSBW     MMGSREMPFN          ; UNLINK PFN FROM FREE OR MODIFY LIST
POPR     #^M<R3>              ; RESTORE SVAPTE
MOVAL    (R5)[R6],R2         ; COMPUTE ADDRESS OF WSL ENTRY
BISL     R11,(R3)            ; SET VALID BIT FOR PTE
ASSUME   PFNSV LOC EQ 0      ; TO USE BISB INSTEAD OF INSV
BISB     #PFNSC_ACTIVE,@W^PFNSAB_STATE[R0]
BICB     #<PFNSM_DELCON!-    ; CLEAR DELETE AND
PFNSM_MODIFY>-              ; MODIFY
@W^PFNSAB_STATE[R0]         ; FLAGS
INCU     @W^PFNSAW_REFCNT[R0] ; RAISE REFERENCE COUNT
CLRW     @W^PFNSAW_SWPVBN[R0] ; INDICATE NO ALTERNATE LOCATION
MOVAL    (R5)[R6],R2         ; COMPUTE ADDRESS OF WSL ENTRY
BBCC     #WSLSV_MODIFY,(R2),50$ ; CLEAR MODIFY BIT FOR WSL
BISB     #PFNSM_MODIFY,@W^PFNSAB_STATE[R0] ; RECORD PAGE AS MODIFIED
MOVW     <R6,@W^PFNSAx WSLX[R0]>,- ; SET WORKING SET LIST INDEX FOR PAG
LONG     OPCODE=MOVZWL,-
IMAGE=SYS_NONPAGED

RSB      ; AND RETURN FOR NEXT PAGE

IS ON THE BAD PAGE LIST. IT HAS THE FOLLOWING POSSIBLE STATES
1) BADPAG BIT SET IN PFNSAB TYPE => BUG CHECK
2) SWPVBN CLEAR => PAGE WRITE ERROR, CORRECT COPY OF MODIFY BIT
   IS THE LOGICAL OR OF THE WSLE BIT AND THE PFN BIT
3) SWPVBN SET => PAGE READ ERROR, SET RDERR STATE.

BBS      #PFNSV_BADPAG,@W^PFNSAB_TYPE[R0],10$ ; ERROR IF BADPAG
PUSHL    R3                  ; SAVE PTE ADDRESS
BSBW     MMGSREMPFN          ; UNLINK PFN FROM THE BAD PAGE LIST
POPR     #^M<R3>              ; RESTORE PTE ADDRESS
MOVAL    (R5)[R6],R2         ; COMPUTE ADDRESS OF WSL ENTRY

```

SUA  
SYM

PTE  
PTE  
PTE  
QEM  
REC  
REL  
REL  
REL  
REL  
RPG  
RSV  
RWS  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SCH  
SET  
SET  
SET  
SGN  
SGN  
SGN  
SGN  
SGN  
SPA  
SUP  
SUA  
SUA  
SUA  
SUA  
SUP  
SUP  
SUP  
SUP  
SUP  
SUP  
SUP  
SUP

```
0000'DF40 B5 0847 2102 TSTW @W^PFNSAW_SWPVBN[R0] : IF SWPVBN SET, THEN PAGE READ ERROR
0D 12 084C 2103 BNEQ 80$ : BRANCH IF PAGE READ ERROR
084E 2104 :
084E 2105 : PAGE WRITE ERROR
084E 2106 :
084E 2107 :
0000'DF40 95 084E 2108 ASSUME PFNSV MODIFY EQ 7
B0 18 0853 2109 TSTB @W^PFNSAB_STATE[R0] : IF PFN MODIFY BIT IS SET
AC 62 08 E2 0853 2110 BGEQ 30$ :
AA 11 0859 2111 BBSS #WLSLV_MODIFY,(R2),30$ : THEN JAM THE WSL ENTRY MODIFY BIT
085B 2112 BRB 30$ : AND CONNECT TO THE PAGE
085B 2113 :
085B 2114 : PAGE READ ERROR
085B 2115 :
085B 2116 :
0000'DF40 14 90 085B 2115 80$: MOVB #<PFNSM_DELCON ! PFNSC_RDERR>,- : SET DELCON
62 0100 8F AA 085D 2116 @W^PFNSAB_STATE[R0] : AND PAGE READ ERROR STATE
AD 11 0861 2117 BICW #<WLSM_MODIFY>,(R2) : CLEAN UP WSL
0866 2118 BRB 45$ : AND LEAVE PTE IN TRANSITION STATE
0868 2119 :
0868 2120 : INSWAP GLOBAL PAGE
0868 2121 :
0868 2122 : GLOBAL:
51 5C 16 00 EF 0868 2123 EXTZV #PTESV_GPTX,#PTESV_GPTX,AP,R1 : GLOBAL PAGE INSWAP
51 0000'DF41 DE 086D 2124 MOVAL @W^MMG$GL_GPTBASE[R1],R1 : GET GLOBAL PAGE TABLE INDEX
52 61 DO 0873 2125 MOVL (R1),R2 : AND CONVERT TO ADDRESS OF GPTE
24 19 0876 2126 BLSS 10$ : PICK UP GLOBAL MASTER PTE
50 52 1D 52 16 E0 0878 2127 BBS #PTESV_TYPO,R2,5$ : BR IF VALID
52 0000'DF40 03 00 EF 087C 2128 EXTZV #PTESV_PFN,#PTESV_PFN,R2,R0 : BR IF GLOBAL SECTION TYPE
18 13 0881 2129 ASSUME PFNSC_FREPAGE EQ 0 : GET PFN OF TRANSITION PAGE
0881 2129 EXTZV #PFNSV_LOC,#PFNSV_LOC,@W^PFNSAB_STATE[R0],R2 : TEST FOR FREE PAGE
0889 2131 BEQL 20$ : YES, REFAULT IT
088B 2132 ASSUME PFNSC_RDINPROG EQ <PFNSC_RDERR + 2> :
088B 2133 CASE R2,2- : DISPATCH ON READ CASES:
088B 2134 55$, - : READ ERROR
088B 2135 4$, - : WRONG STATE
088B 2136 60$ >,- : READ IN PROGRESS
088B 2137 LIMIT=#PFNSC_RDERR :
0057 31 0895 2138 4$: BUG-CHECK ICPAGELOC, FATAL : WRONG STATE - CRASH SYSTEM
0899 2139 5$: BRW 50$ : A BRANCH ASSIST
089C 2140 :
089C 2141 10$:
50 52 15 01D5 30 089C 2142 BSBW RELPAGE : INSWAP WITH VALID GLOBAL PAGE
00 00 EF 089F 2143 EXTZV #PTESV_PFN,#PTESV_PFN,R2,R0 : RELEASE REDUNDANT PAGE
45 11 08A4 2144 BRB 40$ : GET PFN FROM MASTER
08A6 2145 : AND GO SETUP SLAVE PTE
50 FC A9 50 DD 08A6 2146 PUSHL R0 : GLOBAL ON FREE LIST
5A CB 08A8 2147 BICL3 R10,-4(R9),R0 : SAVE MASTER PFN
01C4 30 08AD 2148 BSBW RELPAGE : GET REDUNDANT PFN
01 BA 08B0 2149 POPR #M<R0> : AND RELEASE IT (PRESERVING R1-R3)
61 5B CB 08B2 2150 BICL R11,(R1) : RESTORE MASTER PFN
0A BB 08B5 2151 PUSHR #M<R1,R3> : SET PAGE VALID
F746' 30 08B7 2152 BSBW MMG$REMPFN : SAVE SVAGPTE, SVAPTE
0A BA 08BA 2153 POPR #M<R1,R3> : REMOVE PFN FROM FREELIST
0000'DF40 03 00 07 FO 08BC 2154 INSV #PFNSC_ACTIVE,#PFNSV_LOC,#PFNSV_LOC,@W^PFNSAB_STATE[R0] : RESTORE SVAGPTE, SVAPTE
0000'DF40 B6 08C4 2155 INCW @W^PFNSAW_REFCT[R0] : RAISE REFERENCE COUNT
20 11 08C9 2156 BRB 40$ :
51 51 15 09 EF 08CB 2157 30$: EXTZV #VASV_VPN,#VASV_VPN,R1,R1 : GET VPN OF PAGE TABLE
51 0000'DF41 DO 08D0 2158 MOVL @W^MMG$GL_SPTBASE[R1],R1 : GET PAGE TABLE PTE
```

```
51 51 15 00 EF 08D6 2159 EXTZV #PTESV PFN,#PTESS_PFN,R1,R1 ; EXTRACT PFN
                                08DB 2160 PFN REFERENCE
                                08DB 2161 TSTW <@W*PFNSAx_SHRCNT[R1]>,- ; CHECK FOR FIRST REFERENCE TO PTABL
                                08DB 2162 LONG OPCODE=TSTL,-
                                08DB 2163 IMAGE=SYS_NONPAGED
                                04 12 08E0 2164 BNEQ 358 ; NO
                                08E2 2165 BUG_CHECK GBLPAGSZRO,FATAL ; GLOBAL PAGE SHARE COUNT ZERO
                                08E6 2166 358: PFN REFERENCE
                                08E6 2167 INCW <@W*PFNSAx_SHRCNT[R1]>,- ; RAISE GLOBAL PAGE TABLE SHARE COUN
                                08E6 2168 LONG OPCODE=INCL,-
                                08E6 2169 IMAGE=SYS_NONPAGED
                                08EB 2170 408: PFN REFERENCE
                                08EB 2171 INCW <@W*PFNSAx_SHRCNT[R0]>,- ; RAISE SHARE COUNT FOR GLOBAL PAGE
                                08EB 2172 LONG OPCODE=INCL,-
                                08EB 2173 IMAGE=SYS_NONPAGED
0000'DF40 52 17 FE85 31 08F0 2174 BRW RECONNECT ; RECONNECT AND REFERENCE PAGE TABLE
52 867FFFFF 8F CA 08FB 2175 508: EXTZV #PFNSV BAK,#PFNSS BAK,R2,@W*PFNSAL_BAK[R0] ; SAVE BACKING ADDR
                                52 58 CB 0902 2176 BICL #*C<PTESM_PROT!PTESM_OWN>,R2 ; SAVE PROTECTION AND OWNER FIELDS
                                61 52 50 C9 0905 2177 BISL R1,R2 ; SET PTE VALID
                                0000'DF40 51 D0 0909 2178 BISL3 R0,R2,(R1) ; AND STORE WITH PFN IN GPT
                                0000'DF40 07 90 090F 2179 MOVL R1,@W*PFNSAL_PTE[R0] ; SET SVAGPTE IN PFN DATA BASE
                                0000'DF40 02 90 0915 2180 MOVB #PFNSC_ACTIVE,@W*PFNSAB_STATE[R0] ; SET STATE TO ACTIVE
                                AE 11 091B 2181 MOVB #PFNSC_GLOBAL,@W*PFNSAB_TYPE[R0] ; AND TYPE TO GLOBAL
                                091D 2182 BRB 308 ; NOW GO SETUP SLAVE PTE
                                091D 2183 558: ; PAGE READ ERROR IN GPTE
                                091D 2184 ; THE PFN IN THE GPTE WILL BE DEALLOCATED
                                091D 2185 ; THE GPTE WILL BE ALTERED TO USE THE PFN FROM THE INSWAP IMAGE
                                091D 2186 ; THE DATA BASE WILL BE ADJUSTED AS APPROPRIATE
                                091D 2187
                                091D 2188
0000'DF42 52 FC A9 5A CB 091D 2189 BICL3 R10,-4(R9),R2 ; GET SWAP IMAGE PFN.
                                0000'DF40 D0 0922 2190 MOVL @W*PFNSAL_BAK[R0],@W*PFNSAL_BAK[R2] ; COPY BACKING STORE.
                                0000'DF42 07 90 092B 2191 MOVB #PFNSC_ACTIVE,@W*PFNSAB_STATE[R2] ; SET STATE TO ACTIVE.
                                0000'DF42 02 90 0931 2192 MOVB #PFNSC_GLOBAL,@W*PFNSAB_TYPE[R2] ; SET TYPE TO GLOBAL.
                                0937 2193 PFN REFERENCE = ; COPY SHARE COUNT.
                                0937 2194 MOVW <@W*PFNSAx_SHRCNT[R0],@W*PFNSAx_SHRCNT[R2]>,-
                                0937 2195 LONG OPCODE=MOVL,-
                                0937 2196 IMAGE=SYS_NONPAGED
                                0000'DF40 D4 0940 2197 CLRL @W*PFNSAL_PTE[R0] ; SETUP FOR AND RELEASE
                                012C 30 0945 2198 BSBW RELPAGE ; READ ERROR PFN.
                                0948 2199 ; NOBODY CAN USE IT.
                                50 52 D0 0948 2200 MOVL R2,R0 ; SETUP NEW MASTER PFN.
                                61 15 00 50 F0 094B 2201 INSV R0,#PTESV PFN,#PTESS_PFN,(R1) ; PLANT PFN IN GPTE.
                                0000'DF40 51 D0 0950 2202 MOVL R1,@W*PFNSAL_PTE[R0] ; PLANT PTE IN DATABASE.
                                61 58 CB 0956 2203 BISL R1,(R1) ; MAKE PTE VALID.
                                90 11 0959 2204 BRB 408 ; JOIN COMMON CODE.
                                095B 2205
                                0000'DF40 10 88 095B 2206 608: BISB #PFNSM_COLLISION,@W*PFNSAB_TYPE[R0] ; FLAG COLLISION FOR PAGAREA
                                3C BB 0961 2207 PUSHF #*M<R2,R3,R4,R5> ; SAVE REGS OVER WAIT
                                54 0000'CF D0 0963 2208 MOVL @W*SCH$GL_CURPCB,R4 ; AND SET PCB ADDRESS
                                F695' 30 0968 2209 BSBW SCH$NEULVL ; SET ASTLVL CORRECTLY
                                52 0000'CF 7E 0968 2210 MOVAB @W*SCH$GQ_COLPGWQ,R2 ; GET ADDRESS OF WAIT QUEUE
                                00 DD 0970 2211 PUSHL #0 ; NULL KERNEL MODE PSL
                                F688' 30 0972 2212 BSBW SCH$WAITK ; WAIT WITH NO CALL FRAME
                                0975 2213 SETIPL #IPL$SYNCH ; BLOCK SYSTEM EVENTS
                                50 FC A9 3C BA 0978 2214 POPR #*M<R2,R3,R4,R5> ; RESTORE REGS
                                5A CB 097A 2215 BICL3 R10,-4(R9),R0 ; RESTORE CURRENT PFN
```



```
FEE6 31 097F 2216 BRW GLOBAL ; AND ATTEMPT TO REASSOCIATE PAGE
0982 2217
0982 2218
0982 2219
0982 2220
0982 2221
0982 2222
0982 2223
0982 2224
0982 2225
0982 2226
0982 2227
0982 2228
0982 2229
0982 2230
0982 2231
0982 2232
0982 2233
0982 2234
0982 2235
0982 2236
0982 2237
0982 2238
0982 2239
0982 2240
0982 2241
0982 2242
0982 2243
0982 2244
0982 2245
0982 2246
0982 2247
0982 2248
0982 2249
0982 2250
0982 2251
0982 2252
0982 2253
0982 2254

SET PROPER AST LEVEL
SETASTLVL:
MOVAL PCBSL_ASTQFL(R4),R3 ; GET POINTER TO HEAD OF AST QUEUE
MOVL (R3),R2 ; GET POINTER TO FIRST AST CONTROL BLOCK
CMPL R3,R2 ; IS LIST EMPTY?
BEQL 208 ; YES, DONE
CLRL R0 ; ASSUME KERNEL MODE
MOVB ACBSB_RMOD(R2),R3 ; GET ACTUAL MODE
BLSS 108 ; BR IF SPECIAL KERNEL AST
EXTZV #ACBSV_MODE,#ACBSB_MODE,R3,R0 ; GET ACCESS MODE
BICB3 PCBSB_ASTACT(R4),PCBSB_ASTEN(R4),R1 ; CHECK FOR DELIVERABILITY
BBC R0,R1,208 ; BR IF NOT PRESENTLY DELIVERABLE
MOVB R0,PHDSB_ASTLVL(R5) ; SET AST LEVEL FOR PROCESS
BISL #<<10PCBSV_RES>>!<10PCBSV_INQUAN>>,PCBSL_STS(R4) ; MARK PROCESS RESID
MOVW SCHSGW_QUAN,PHDSW_QUANT(R5) ; AND GIVE NEW QUANTUM
MOVZBL PCBSB_PRI(R4),R0 ; GET CURRENT PRIORITY OF PROCESS
SUBB3 R0,#3T,R1 ; COMPUTE EXTERNAL PRIORITY FOR COMPARE
CMPB R1,W*SYSSGB_DEFPRI ; IS THIS A 'CRUNCHER'?
BGTR 308 ; NO, CONTINUE
MOVL EXESGB_SYSTIME,R1 ; GET CURRENT TIME IN APPROX. 10MS UNITS
ADDL3 W*SCHSGL_SWPRATE,R1,W*SWPSGL_SWTIME ; SET NEW CRUNCHER INTERVAL
BSBW SCHSCHSEP ; CHANGE TO RESIDENT COMPUTE
SWAPRETRY: ; RETRY SWAP SCHEDULING
MOVL W*SCHSGL_CURPCB,R4 ; GET PCB ADDRESS
BBSI1 #PCBSV_WAKEPEN,PCBSL_STS(R4),208 ; SET TO CANCEL HIBER
208:
.DSABL LSB
SWAPEXIT: ; EXIT SWAPPER
BBC S*#SCHSV_SIP,W*SCHSGB_SIP,108 ; CLEAR SWAP IN PROGRESS
108:
SWAPEXITA: ; ALTERNATE EXIT, LEAVING SIP SET
POPR #*M<R6,R7,R8,R9,R10,R11,AP,FP> ; RESTORE REGISTERS
SETIPL #0 ; DROP IPL
RSB
```

```

.SBTTL  FILLPHD - FILL SPT ENTRIES TO MAP PHD

**
FUNCTIONAL DESCRIPTION:
  FILLPHD SETS THE PTE ENTRIES FOR THE PROCESS HEADER INTO THE
  SPT.

CALLING SEQUENCE:
  BSB/JSB FILLPHD

INPUT PARAMETERS:
  R3 - POINTER TO FIRST SPT ENTRY FOR PHD
  R9 - ADDRESS OF SWAPPER MAP ENTRY TO BE MOVED TO SPT
  R10 - PTESC_ERKW!PTESM_VALID!PTESM_MODIFY

OUTPUT PARAMETERS:
  R5 - ZERO
  R6 - DESTROYED
  R9 - UPDATED
  R11 - DESTROYED
  AP - DESTROYED
  FP - DESTROYED

--

FILLPHD:

      CLRL    R5                ; SET PHD ADDRESS TO SWAPPER PO SPACE
      INVALID                ; TO SEE CORRECT PROCESS HEADER IN SWAPPER P
      CLRL    R11              ; INIT HEADER PAGE INDEX
      MOVZWL  PHDSW_EMPTYPG(R5),R6 ; GET COUNT OF EMPTY PAGES
      ASHL    #9,R6,R6         ; CONVERT TO BYTE OFFSET
      MOVL    PHDSL_WSLX(R5),AP ; FORM BASE ADDRESS FOR WSLX
      MOVAL   (R5)[AP],AP      ; SAVE VECTOR FOR PHD
      MOVL    PHDSL_BAK(R5),FP  ; FORM BASE ADDRESS FOR BACKING STORE ADDRESS
      MOVAL   (R5)[FP],FP      ; VECTOR
      SUBL    R6,AP            ; ACCOUNT FOR EMPTY PAGES
      SUBL    R6,FP            ; BY SUBTRACTING THEIR SPACE
      MOVL    W^SWP$GL_BSLOTSZ,R6 ; SET ITERATION COUNT FOR ENTIRE HEADER
108:  MOVL    (FP)+,(R3)+        ; SET BACKUP FORM OF PTE IN SPT SLOT
      BGEQ    308              ; DONE IF NOT VALID
      BICL3   R10,(R9)+,R0     ; GET PAGE FROM SWAPPER MAP
      MOVAL   -(R3),@W^PFNSAL_PTE[R0] ; SET PTE BACK POINTER
      EXTZV   #PFNSV_BAK,#PFNSS_BAK,(R3),R1 ; ISOLATE BACKING STORE ADDRESS
      INSV    PHDSB_PAGFI(L(R5),#PFNSV_PGFLX,#PFNSS_PGFLX,R1 ; ADD FILE NUMBER
      MOVL    R1,@W^PFNSAL_BAK[R0] ; SAVE IN PFN DATA BASE
      PFN REFERENCE -
      MOVW    <(AP)[R11],@W^PFNSAx_WSLX[R0]>,- ; SAVE WORKING SET LIST INDE
      LONG OPCODE=MOVZWL,-
      IMAGE=SYS_NONPAGED
      BISL3   R0,R10,(R3)+      ; SET VALID PTE FOR PAGE
      MOVW    #<PFNSC_ACTIVE!PFNSM_MODIFY>,@W^PFNSAB_STATE[R0] ; MARK PAGE ACTIVE
      MOVW    #PFNSC_PPGTBL,@W^PFNSAB_TYPE[R0] ; STORE TYPE IN PFN DATA BAS
308:  AOBLS    R6,R11,108        ; FILL ENTIRE PROCESS HEADER
      RSB

```

```
0A57 2313 .SBTTL RELINIT - INITIALIZE REGISTERS FOR PAGE RELEASE LOOP
0A57 2314
0A57 2315 :++
0A57 2316 : FUNCTIONAL DESCRIPTION:
0A57 2317 : RELINIT SETS UP REGISTERS FOR THE PAGE RELEASE LOOPS FOLLOWING
0A57 2318 : OUTSWAP I/O OPERATIONS.
0A57 2319 :
0A57 2320 : CALLING SEQUENCE:
0A57 2321 : BSB/JSB RELINIT
0A57 2322 :
0A57 2323 : INPUT PARAMETERS:
0A57 2324 : NONE
0A57 2325 :
0A57 2326 : OUTPUT PARAMETERS:
0A57 2327 : R0 - 0
0A57 2328 : R4 - OUT SWAP PCB ADDRESS (OSWPPCB)
0A57 2329 : R7 - PAGE COUNT TO RELEASE
0A57 2330 : R9 - BASE ADDRESS FOR SWAPPER MAP (SWPSAL_MAP)
0A57 2331 : R10 - PTESB ERKW!PTESM_VALID!PTESM_MODIFY
0A57 2332 : R11 - BASE ADDRESS FOR SWAPPER MAP (SWPSAL_MAP)
0A57 2333 :
0A57 2334 : --
0A57 2335 :
0A57 2336 RELINIT:
54 0014'CF DO 0A57 2337 : MOVL W*OSWPPCB,R4 : RELEASE LOOP INITIALIZATION
57 0012'CF SC 0A5C 2338 : MOVZWL W*OSWPPGS,R7 : GET PCB ADDRESS OF OUT SWAP PROCESS
0A61 2339 : BRB OSINIT : AND PAGE COUNT FOR RELEASE LOOP
: : FALL INTO OSINIT
```



```
0A61 2342 .SBTTL OSINIT - OUTSWAP SCAN REGISTER INITIALIZATION
0A61 2343
0A61 2344 :++
0A61 2345 : FUNCTIONAL DESCRIPTION:
0A61 2346 : OSINIT SETS UP REGISTERS FOR PAGE TABLE SCANS REQUIRED DURING
0A61 2347 : OUTSWAPPING.
0A61 2348 :
0A61 2349 : INPUT PARAMETERS:
0A61 2350 : NONE
0A61 2351 :
0A61 2352 : OUTPUT PARAMETERS:
0A61 2353 : R9 - BASE ADDRESS OF SWAPPER MAP (SWPSAL_MAP)
0A61 2354 : R10 - PTESC_ERKW!PTESM_VALID
0A61 2355 : R11 - BASE ADDRESS OF SWAPPER MAP (SWPSAL_MAP)
0A61 2356 :
0A61 2357 :--
0A61 2358
0A61 2359 OSINIT:
59 0000'DF DE 0A61 2360 MOVAL @W^SWPSGL_MAP,R9 ; SET BASE OF SWAPPER MAP
SB 59 DO 0A66 2361 MOVL R9,R11 ; AND MAKE REFERENCE COPY
SA B4000000 8F DO 0A69 2362 MOVL #<PTESC_ERKW!PTESM_VALID!PTESM_MODIFY>,R10 ; MASK TO VALIDATE SWAP P
OS 0A70 2363 RSB ; RETURN
```

```
0A71 2366 .SBTTL RELPAGE - RELEASE DUPLICATE PAGE
0A71 2367
0A71 2368 :++
0A71 2369 : FUNCTIONAL DESCRIPTION:
0A71 2370 : RELPAGE RELEASES A PHYSICAL PAGE WHICH DUPLICATES A PAGE ALREADY
0A71 2371 : PRESENT FOR THE PROCESS. THIS SITUATION CAN ARISE DUE TO A PARTIAL
0A71 2372 : INSWAP OR A GLOBAL PAGE WHICH IS ALREADY PRESENT.
0A71 2373 :
0A71 2374 : CALLING SEQUENCE:
0A71 2375 : BSB/JSB RELPAGE
0A71 2376 :
0A71 2377 : INPUT PARAMETERS:
0A71 2378 : R0 - PFN TO RELEASE
0A71 2379 : R3 - SVA OF PTE (RELDELPAGE ONLY)
0A71 2380 :
0A71 2381 : OUTPUT PARAMETERS:
0A71 2382 : R1 - PRESERVED (RELPAGE ONLY)
0A71 2383 : R2 - PRESERVED (RELPAGE ONLY)
0A71 2384 : R3 - PRESERVED (RELPAGE ONLY)
0A71 2385 :
0A71 2386 :--
0A71 2387
0A71 2388 RELDELPAGE:
0A71 2389 BSBW MMGSDELCONPFN : RELEASE PAGE THROUGH DELCONPFN
0A71 2390 RELPAGE: : DELETE PAGE CONTENT AND INIT PFN DATA
0A71 2391 : RELEASE PAGE
0A71 2392 : PRESERVE REGISTERS
0A71 2393 : INIT PFN DATA FOR RELEASE
0A71 2394 : ZERO REFERENCE COUNT
0A71 2395 :
0A71 2396 : INDICATE FREELIST
0A71 2397 : RELEASE PFN TO HEAD OF FREE LIST
0A71 2398 : RESTORE REGISTERS
0A71 2399 : AND RETURN TO CALLER

F58C' 30 0A71 2388 RELDELPAGE:
OE BB 0A71 2389 BSBW MMGSDELCONPFN
0000'DF40 94 0A74 2390 RELPAGE:
0000'DF40 B4 0A74 2391 : RELEASE PAGE
52 D4 0A74 2392 : PRESERVE REGISTERS
F57B' 30 0A76 2393 : INIT PFN DATA FOR RELEASE
OE BA 0A7B 2394 : ZERO REFERENCE COUNT
05 0A80 2395 :
0A80 2396 : INDICATE FREELIST
0A82 2397 : RELEASE PFN TO HEAD OF FREE LIST
0A85 2398 : RESTORE REGISTERS
0A87 2399 : AND RETURN TO CALLER

PUSHR #*M<R1,R2,R3>
CLRB @W*PFNSAB_STATE[R0]
CLRW @W*PFNSAW_REFcnt[R0]
ASSUME PFNSC_FREPAGLST EQ 0
CLRL R2
BSBW MMGSINSPFNH
POPR #*M<R1,R2,R3>
RSB
```

```
0A88 2401      .SBTTL  SWPREAD/SWPWRITE - SWAPPER I/O ROUTINES
0A88 2402
0A88 2403      :++
0A88 2404      : FUNCTIONAL DESCRIPTION:
0A88 2405      : SWPREAD AND SWPWRITE PERFORM THE DETAIL WORK REQUIRED TO READ
0A88 2406      : OR WRITE A SET OF CONTIGUOUS PAGES IN A WORKING SET SWAP IMAGE.
0A88 2407      : THE CALL TO EITHER SWPREAD OR SWPWRITE IS ACTUALLY A CO-ROUTINE
0A88 2408      : CALL WHICH RETURNS ONLY AFTER ALL SEGMENTS OF THE I/O OPERATION
0A88 2409      : HAVE BEEN PERFORMED. THIS RETURN IS EFFECTED BY A SPECIAL KERNEL
0A88 2410      : AST.
0A88 2411
0A88 2412      : CALLING SEQUENCE:
0A88 2413      : BSB/JSB SWPREAD/SWPWRITE
0A88 2414
0A88 2415      : INPUT PARAMETERS:
0A88 2416      : R0 - SWAP FILE INDEX
0A88 2417      : R2 - WSSWP FORM DISK ADDRESS
0A88 2418      : R3 - SYSTEM VIRTUAL ADDRESS OF PTE
0A88 2419      : R4 - PAGE COUNT
0A88 2420
0A88 2421      : 00(SP) - RETURN ADDRESS AFTER I/O COMPLETION
0A88 2422      : 04(SP) - SAVED R6
0A88 2423      : 08(SP) - SAVED R7
0A88 2424      : 12(SP) - SAVED R8
0A88 2425      : 16(SP) - SAVED R9
0A88 2426      : 20(SP) - SAVED R10
0A88 2427      : 24(SP) - SAVED R11
0A88 2428      : 28(SP) - SAVED AP
0A88 2429      : 32(SP) - SAVED FP
0A88 2430      : 36(SP) - SAVED IPL
0A88 2431      : 40(SP) - RETURN TO PREVIOUS THREAD
0A88 2432
0A88 2433      : IMPLICIT INPUTS:
0A88 2434      : PAGE FILE TABLE ENTRY (PFL) SELECTED BY WSSWP INPUT
0A88 2435
0A88 2436      : OUTPUT PARAMETERS:
0A88 2437      : R0 - COMPLETION STATUS OF I/O OPERATION
0A88 2438
0A88 2439      :--
0A88 2440
0A88 2441      .ENABL  LSB
0A88 2442  SWPREAD:  PUSHAB  W*EXESBLDPKTSWPR      : SWAP READ INITIATION
0A88 2443      BRB      10$      : SET ADDRESS OF BUILD PACKET ROUTINE
0A88 2444
0A88 2445  SWPWRITE:  PUSHAB  W*EXESBLDPKTSWPW      : SWAP WRITE INITIATION
0A88 2446      10$:  MOVAB   W*IOROUTINE,R1      : SET ADDRESS OF BUILD PACKET ROUTINE
0A88 2447      MOVQ   (SP)+,(R1)+      : ADDRESS OF I/O DATA
0A88 2448      POPR   #*M<R6,R7,R8,R9,R10,R11,AP,FP>: SAVE I/O END ACTION ADDRESS
0A88 2449      EXTZV  #24,#8,R2,R0      : RESTORE REGISTERS OTHER THAN STANDAR
0A88 2450      15$:  MOVL    @W*MMG$GL_PAGSWPVC[R0],R0; GET SWAP FILE INDEX
0A88 2451      MOVZBL #127,R5      : GET BASE ADDRESS OF PAGE FILE TABLE
0A88 2452      CMPL   R4,R5      : SET I/O SIZE
0A88 2453      BGTR   20$      : COMPARE REMAINING PGCNT WITH MAX TRANSFER
0A88 2454      MOVL   R4,R5      : USE MAXIMUM TRANSFER
0A88 2455      20$:  ADDL3   R5,R2,(R1)+      : SET TRANSFER TO REMAINING PAGES
0A88 2456      MOVAL  (R3)[R5],(R1)+      : SAVE UPDATED DISK ADDRESS
0A88 2457      : AND UPDATED SAVPTE
```

0000'CF	9F	0A88	2442
04	11	0A8C	2443
		0A8E	2444
51	0000'CF	9F	0A8E
	81	8E	7D
	3FC0	8F	BA
50	52	08	18
		EF	0A9E
50	50	0000'DF40	D0
	55	7F	8F
		55	54
			03
	55	54	D0
81	52	55	C1
	81	6345	DE



61	54	55	A3	OABD	2458	SETIPL	#0	:	DROP IPL
		53	DD	OACO	2459	SUBW3	R5,R4,(R1)	:	SAVE REMAINING PAGE COUNT
		OC	DD	OAC4	2460	PUSHL	R3	:	SAVE SVAPTE
7E	55	09	9C	OAC6	2461	PUSHL	PFL\$L_WINDOW(R0)	:	GET WINDOW ADDRESS
52	18	00	EF	OAC9	2462	ROTL	#9,R5--(SP)	:	CONVERT PAGES TO BYTE COUNT
	6E	10	CO	OACD	2463	EXTZV	#0,#24,R2--(SP)	:	AND ISOLATE BLOCK NUMBER
		A0	CO	OAD2	2464	ADDL	PFL\$L_VBN(R0),(SP)	:	ADD BASE VBN
54	0000	CF	DO	OAD6	2465	MOVL	W*SCH\$GL_CURPCB,R4	:	SET PCB ADDRESS
55	0000	DF	OF	OADB	2466	REMQUE	W*IOC\$GE_IRPFL,R5	:	GET A PACKET IF POSSIBLE
		06	1C	OAE0	2467	BVC	30\$	:	BR IF ONE AVAILABLE
		F51B	30	OAE2	2468	BSBW	EXE\$ALLOCIRP	:	ALLOCATE ONE THE LONG WAY
	55	52	DO	OAE5	2469	MOVL	R2,R5	:	SET PACKET ADDRESS IN PROPER REGISTER
23 A5	14 A5	FB AF	9E	OAE8	2470	MOVAB	B*IODONE,IRP\$L_ASTPRM(R5)	:	SET ADDRESS FOR COMPLETION
	1F	0000 CF	83	OAED	2471	SUBB3	W*SWP\$GB_PRI0,#31,IRP\$B_PRI(R5)	:	SET PRIORITY FOR TRANSFER
		OF	BA	OAF4	2472	POPR	#*M<R0,RT,R2,R3>	:	RESTORE VBN,BYTECNT,WINDOW,SVAPTE
		0000 DF	16	OAF6	2473	JSB	W*IOROUTINE	:	CALL READ OR WRITE ROUTINE
			05	OAF8	2474	RSB		:	AND RETURN TO ORIGINAL CALLER
				OAFB	2475			:	
				OAFB	2476			:	
	38 A5		DD	OAFB	2477	PUSHL	IRP\$L_MEDIA(R5)	:	CONTINUATION CALLED AS KERNEL AST
	50	55	DO	OAFE	2478	MOVL	R5,R0	:	SAVE COMPLETION STATUS
		F4FC	30	OB01	2479	BSBW	EXE\$DEANONPAGED	:	SET PACKET ADDRESS FOR RELEASE
	50	8E	DO	OB04	2480	MOVL	(SP)+,R0	:	AND RELEASE IT
				OB07	2481	SETIPL	#IPL\$ SYNCH	:	RESTORE STATUS
	11 50		E9	OB0A	2482	BLBC	R0,60\$	:	BLOCK SYSTEM EVENTS
51	0008 CF		9E	OB0D	2483	MOVAB	W*RWSSWP,R1	:	EXIT IF ERROR
	52	61	7D	OB12	2484	MOVQ	(R1),R2	:	GET ADDRESS OF REMAINING TRANSFER PARAMS
	54	0B A1	3C	OB15	2485	MOVZWL	B* <RPGCNT-RWSSWP>(R1),R4	:	RESTORE WSSWP,SVAPTE TO R2,R3
		03	13	OB19	2486	BEQL	60\$	:	AND REMAINING PAGE COUNT
		FF80	31	OB1B	2487	BRW	15\$	:	DONE IF NO MORE PAGES REMAIN
	3FC0 8F		BB	OB1E	2488	PUSHR	#*M<R6,R7,R8,R9,R10,R11,AP,FP>	:	CONTINUE IF MORE PAGES REMAIN
	0004 DF		17	OB22	2489	JMP	W*IOEA	:	SAVE NON-STANDARD REGISTERS
				OB26	2490			:	AND CONTINUE SWAP
				OB26	2491			:	
				OB26	2492			:	
						.DSABL	LSB	:	
						.END		:	

SWAPPER  
Symbol table

## WORKING SET SWAPPER

N 6

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1Page 54  
(28)SYS  
V04

```
SSARGS      = 00000005
$ST1        = 00000018
...PFN      = 00000A3A R    05
ACBSB_RMOD  = 00000008
ACBSS_MODE  = 00000002
ACBSV_MODE  = 00000000
BALANCE     = 00000030 R    05
BDLSL_SYSDLOG ***** X    04
BDLSS_CRELNMTMLST ***** X    04
BUGS_APTREFHIGH ***** X    05
BUGS_APTWRERR ***** X    05
BUGS_GBLPAGSZRO ***** X    05
BUGS_ICPAGELOC ***** X    05
BUGS_INSNFREPAGE ***** X    05
BUGS_INSSWPFIL ***** X    05
BUGS_INSWAPERR ***** X    05
BUGS_IVWSETLIST ***** X    05
BUGS_OUTSWPERR ***** X    05
BUGS_QUEUEEMPTY ***** X    05
BUGS_ZEROPAGE ***** X    05
COPYSHELL   = 000005DF R    05
CRELNMS_ACMODE = 00000010
CRELNMS_ATTR  = 00000004
CRELNMS_ITMLST = 00000014
CRELNMS_LOGNAM = 0000000C
CRELNMS_NARGS = 00000005
CRELNMS_TABNAM = 00000008
DELCON       = 0000051F R    05
DELPHD       = 00000404 R R R 05
DIRECTORIES_ARG = 0000032D R R R 04
DIRECTORIES_LIST = 000001DD R    04
DYNBS_LNM    = 00000040
DYNBS_ORB    = 00000049
DYNBS_PCB    = 0000000C
DYNBS_RSHT   = 00000038
EXESALOCIRP ***** X    05
EXESALOPAGED ***** X    04
EXESBLDPKTSWPR ***** X    05
EXESBLDPKTSWPU ***** X    05
EXESDEANONPAGED ***** X    05
EXESDEANONPGDSIZ ***** X    04
EXESGL_PAGED ***** X    04
EXESGL_PFATIM ***** X    05
EXESGQ_SYSTIME ***** X    05
EXESPODERAST ***** X    05
EXESSWAPINIT = 000004AC RG    04
EXEC_MODE    = 000001C9 R    04
FILE_DEV_EXEC_ARG = 00000345 R    04
FILE_DEV_EXEC_LIST = 00000239 R    04
FILE_DEV_SUPER_ARG = 0000035D R    04
FILE_DEV_SUPER_LIST = 00000205 R    04
FILLPHD     = 000009EF R    05
GBLDROP     = 00000462 R    05
GBLRESET    = 000004B1 R    05
GBLTRANS    = 0000045A R    05
GBLVALID    = 0000049E R    05
GBLWRTTRANS = 0000045A R    05
```

```
GBLWRTVALID 000004CF R    05
GLOBAL      00000868 R R R 05
IMGDESC     00000000 R R R 04
INSWAP      00000535 R R R 05
IOC$GL_IRPFL ***** X    05
IODONE      00000AFB R R R 05
IOEA        00000004 R R R 02
IOROUTINE   00000000 R R R 02
IPL$_ASTDEL = 00000002
IPL$_SYNCH  = 00000008
IRPSB_PRI   = 00000023
IRPSL_ASTPRM = 00000014
IRPSL_MEDIA = 00000038
KERNEL_MODE 000001CD R    04
LNMSAL_HASHTBL ***** X    04
LNMSGH_HTBLSIZ ***** X    04
LNMSHASH     ***** X    04
LNMSINSLOGTAB ***** X    04
LNMSM_NO_ALIAS = 00000001
LNMSM_TERMINAL = 00000200
LNMS$SYSTEM_DIRECTORY 00000000 RG    03
LNMS_ATTRIBUTES = 00000003
LNMS_STRING  = 00000002
LNMS$B_ACMODE = 00000008
LNMS$B_FLAGS = 00000010
LNMS$B_TYPE  = 0000000A
LNMS$B_BLINK = 00000004
LNMS$B_FLINK = 00000000
LNMS$B_TABLE = 0000000C
LNMS$M_NODELETE = 00000010
LNMS$M_NO_ALIAS = 00000001
LNMS$M_TABLE  = 00000008
LNMS$T_NAME   = 00000011
LNMS$W_SIZE   = 00000008
LNMS$H$B_TYPE = 0000000A
LNMS$H$B_BUCKET = 0000000C
LNMS$H$B_BUCKET = 0000000C
LNMS$H$B_MASK = 00000000
LNMS$H$B_SIZE = 00000008
LNMS$H$B_FLAGS = 00000000
LNMS$H$B_LENGTH = 00000025
LNMS$H$B_BYTES = 00000021
LNMS$H$B_BYTESLM = 0000001D
LNMS$H$B_CHILD = 00000011
LNMS$H$B_HASH  = 00000001
LNMS$H$B_NAME  = 00000009
LNMS$H$B_ORB   = 00000005
LNMS$H$B_PARENT = 0000000D
LNMS$H$B_QTABLE = 00000019
LNMS$H$B_SIBLING = 00000015
LNMS$H$B_DIRECTORY = 00000002
LNMS$H$B_SHAREABLE = 00000001
LNMS$H$B_SYSTEM = 00000008
LNMS$B_FLAGS   = 00000000
LNMS$B_INDEX   = 00000001
LNMS$C_TABLE   = FFFFFFFF82
LNMS$M_TERMINAL = 00000002
```

SWAPPER  
Symbol table

## WORKING SET SWAPPER

B 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1Page 55  
(28)SYS  
V04

LNMXSM_XEND	=	00000004		
LNMXST_XLATION	=	00000004		
LNMXSW_HASH	=	00000002		
LNMX_DIRECTORY_DESC		00000020	R	04
LNMX_FILE_DEV_DESC		00000037	R	04
LNMX_GROUP		00000140	R	04
LNMX_GROUP_LENGTH	=	00000009		
LNMX_JOB		00000156	R	04
LNMX_JOB_LENGTH	=	00000007		
LNMX_NO_ALIAS		00000105	R	04
LNMX_PERMANENT_MAILBOX_DESC		00000048	R	04
LNMX_PROCESS		00000150	R	04
LNMX_PROCESS_DIRECTORY		00000168	R	04
LNMX_PROCESS_DIRECTORY_LENGTH	=	00000015		
LNMX_PROCESS_LENGTH	=	00000008		
LNMX_SYSTEM		00000170	R	04
LNMX_SYSTEM_DESC		00000068	R	04
LNMX_SYSTEM_DIRECTORY		00000187	R	04
LNMX_SYSTEM_DIRECTORY_DESC		00000070	R	04
LNMX_SYSTEM_DIRECTORY_LENGTH	=	00000014		
LNMX_SYSTEM_DIR_LNMTH		0000002B	RG	03
LNMX_SYSTEM_DIR_ORB		00000058	R	03
LNMX_SYSTEM_LENGTH	=	0000000A		
LNMX_SYSTEM_TABLE		0000019B	R	04
LNMX_SYSTEM_TABLE_LENGTH	=	00000010		
LNMX_SYS_DIR_ORB_SIZ	=	00000068		
LNMX_SYS_DIR_SIZ	=	000000C0		
LNMX_TEMPORARY_MAILBOX_DESC		00000078	R	04
LOG_GROUP		000001AB	R	04
LOG_GROUP_LENGTH	=	00000009		
LOG_G_ARG		00000375	R	04
LOG_G_DESC		00000095	R	04
LOG_G_LIST		00000249	R	04
LOG_PROCESS		000001B4	R	04
LOG_PROCESS_LENGTH	=	0000000B		
LOG_P_ARG		00000380	R	04
LOG_P_DESC		00000090	R	04
LOG_P_LIST		00000259	R	04
LOG_SYSTEM		000001BF	R	04
LOG_SYSTEM_LENGTH	=	0000000A		
LOG_S_ARG		000003A5	R	04
LOG_S_DESC		000000A5	R	04
LOG_S_LIST		00000275	R	04
LOOP		00000000	R	05
MMGSALLOCPFN		*****	X	05
MMGSAL_SYSPCB		*****	X	05
MMGSALCPAGFIL		*****	X	05
MMGSDECPTRF		*****	X	05
MMGSDELCONPFN		*****	X	05
MMGSDELWSLEX		*****	X	05
MMGSGB_FREUFLGS		*****	X	05
MMGSGL_GPTBASE		*****	X	05
MMGSGL_PAGSUPVC		*****	X	05
MMGSGL_SPTBASE		*****	X	05
MMGSINSPFNM		*****	X	05
MMGSINSPFNT		*****	X	05
MMGSIOLOCKPAG		*****	X	05

MMGSM_NOWAIT	*****	X	05
MMGSREFCNTNEG	*****	X	05
MMGSRELPFN	*****	X	05
MMGSREMPFN	*****	X	05
MMGSSHRCNTNEG	*****	X	05
MMGSSVAPTECHK	*****	X	05
MMGSUNLOCK	*****	X	05
MMGSWRTHFYFAG	*****	X	05
NOTSHELL	00000689	R	05
NOTVALID	00000231	R	05
NTYP1	000007C2	R	05
OPS_CMPL	= 000000D1		
OPS_CMPW	= 000000B1		
OPS_CVTLW	= 000000F7		
OPS_DECL	= 000000D7		
OPS_DECW	= 000000B7		
OPS_INCL	= 000000D6		
OPS_INCW	= 000000B6		
OPS_MOVL	= 000000D0		
OPS_MOVW	= 000000B0		
OPS_MOVZWL	= 0000003C		
OPS_TSTL	= 000000D5		
OPS_TSTW	= 000000B5		
ORBSB_FLAGS	= 0000000B		
ORBSB_TYPE	= 0000000A		
ORBSK_LENGTH	= 00000058		
ORBSL_ACL_COUNT	= 00000028		
ORBSL_ACL_DESC	= 0000002C		
ORBSL_ACL_MUTEX	= 00000004		
ORBSL_GRP_PROT	= 00000020		
ORBSL_OWNER	= 00000000		
ORBSL_OWN_PROT	= 0000001C		
ORBSL_SYS_PROT	= 00000018		
ORBSL_WOR_PROT	= 00000024		
ORBSQ_MODE_PROT	= 00000010		
ORBSR_MAX_CLASS	= 00000044		
ORBSR_MIN_CLASS	= 00000030		
ORBS_S_MAX_CLASS	= 00000014		
ORBS_S_MIN_CLASS	= 00000014		
ORBSW_REFCOUNT	= 0000000E		
ORBSW_SIZE	= 00000008		
OSDISPATCH	00000237	R	05
OSINIT	00000A61	R	05
OSWPEXIT	00000401	R	05
OSWPPCB	00000014	R	02
OSWPPGS	00000012	R	02
OUTSWAP	00000116	R	05
OWSLOOP	00000215	R	05
P1SYSVECTORS	*****	X	04
PCBSB_ASTACK	= 0000000C		
PCBSB_ASTACK	= 0000000D		
PCBSB_PRI	= 0000000B		
PCBSB_TYPE	= 0000000A		
PCBSL_ASTACK	= 00000010		
PCBSL_PHD	= 0000006C		
PCBSL_PHYPCB	= 00000018		
PCBSL_PID	= 00000060		



SWAPPER  
Symbol table

## WORKING SET SWAPPER

C 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1Page 56  
(28)SYS  
V04

PCBSL_STS	= 00000024		
PCBSL_WSSWP	= 00000020		
PCBSV_INQUAN	= 00000003		
PCBSV_PHDRES	= 00000012		
PCBSV_RES	= 00000000		
PCBSV_WAKPEN	= 0000000C		
PCBSW_APTCNT	= 00000030		
PCBSW_GPGCNT	= 00000034		
PCBSW_PPGCNT	= 00000036		
PERMANENT_MAILBOX_ARG	000003BD	R	04
PERMANENT_MAILBOX_LIST	00000239	R	04
PFLSL_VBN	= 00000010		
PFLSL_WINDOW	= 0000000C		
PFNSAB_STATE	*****	X	05
PFNSAB_TYPE	*****	X	05
PFNSAL_BAK	*****	X	05
PFNSAL_HEAD	*****	X	05
PFNSAL_PTE	*****	X	05
PFNSAW_REFCNT	*****	X	05
PFNSAW_SWPVB	*****	X	05
PFNSAX_FLINK	*****	X	05
PFNSAX_SHRCNT	*****	X	05
PFNSAX_WSLX	*****	X	05
PFNSC_ACTIVE	= 00000007		
PFNSC_BADPAGLST	= 00000002		
PFNSC_FREPAGLST	= 00000000		
PFNSC_GBLWRT	= 00000003		
PFNSC_GLOBAL	= 00000002		
PFNSC_GPGTBL	= 00000005		
PFNSC_MFY PAGLST	= 00000001		
PFNSC_PPGTBL	= 00000004		
PFNSC_PROCESS	= 00000000		
PFNSC_RDERR	= 00000004		
PFNSC_RDINPROG	= 00000006		
PFNSC_RELPEND	= 00000003		
PFNSC_SYSTEM	= 00000001		
PFNSC_WRTINPROG	= 00000005		
PFNSM_COLLISION	= 00000010		
PFNSM_DELCON	= 00000010		
PFNSM_MODIFY	= 00000080		
PFNSS_BAK	= 00000017		
PFNSS_LOC	= 00000003		
PFNSS_PAGTYP	= 00000003		
PFNSS_PGFLX	= 00000008		
PFNSV_BADPAG	= 00000005		
PFNSV_BAK	= 00000000		
PFNSV_LOC	= 00000000		
PFNSV_MODIFY	= 00000007		
PFNSV_PAGTYP	= 00000000		
PFNSV_PGFLX	= 00000018		
PHDSB_ASTLVL	= 000000CF		
PHDSB_PAGFIL	= 0000001F		
PHDSL_BAK	= 00000044		
PHDSL_POBR	= 000000C8		
PHDSL_P1BR	= 000000D0		
PHDSL_PCB	= 00000078		
PHDSL_WSLX	= 00000048		

PHDSV_NOACCVIO	= 00000003		
PHDSW_EMPTYPG	= 00000008		
PHDSW_FLAGS	= 00000036		
PHDSW_PHVINDE	= 00000042		
PHDSW_PTCNTACT	= 00000070		
PHDSW_QUANT	= 0000003C		
PHDSW_SWAPSIZE	= 00000052		
PHDSW_WSLAST	= 00000012		
PHDSW_WSLIST	= 00000008		
PHVSGC_PIXBAS	*****	X	05
PHVSGC_REFCBAS	*****	X	05
PPGTBLTRANS	0000052A	R	05
PPGTBLVALID	0000052A	R	05
PQLSAB_SYSPQL	00000465	R	04
PQLSC_SYSPQLLEN	= 00000046	G	
PQLS_ASTLM	= 00000001		
PQLS_BIOLM	= 00000002		
PQLS_BYTLM	= 00000003		
PQLS_CPULM	= 00000004		
PQLS_DIOLM	= 00000005		
PQLS_ENQLM	= 0000000C		
PQLS_FILLM	= 00000006		
PQLS_JTQUOTA	= 0000000E		
PQLS_LISTEND	= 00000000		
PQLS_PGFLQUOTA	= 00000007		
PQLS_PRCLM	= 00000008		
PQLS_TQELM	= 00000009		
PQLS_WSDEFAULT	= 0000000B		
PQLS_WSEXTENT	= 0000000D		
PQLS_WSQUOTA	= 0000000A		
PR\$_TPL	= 00000012		
PR\$_TBIA	= 00000039		
PRCSM_NOACNT	= 00000008		
PRCSM_SSRWAIT	= 00000001		
PROCDROP	0000047B	R	05
PROCTRANS	000004DE	R	05
PROCVALID	000004F0	R	05
PROCWRT	00000258	R	05
PSLSC_EXEC	= 00000001		
PSLSC_KERNEL	= 00000000		
PSLSC_SUPER	= 00000002		
PTESC_ERKW	= 30000000		
PTESC_URKW	= 70000000		
PTESM_MODIFY	= 04000000		
PTESM_OW	= 01800000		
PTESM_PFN	= 001FFFFF		
PTESM_PROT	= 78000000		
PTESM_TYPO	= 00400000		
PTESM_TYPI	= 04000000		
PTESM_VALID	= 80000000		
PTESS_GPTX	= 00000016		
PTESS_PFN	= 00000015		
PTESS_PGFLVB	= 00000016		
PTESV_GPTX	= 00000000		
PTESV_MODIFY	= 0000001A		
PTESV_PFN	= 00000000		
PTESV_PGFLVB	= 00000000		

SWAPPER  
Symbol table

## WORKING SET SWAPPER

D 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1Page 57  
(28)SYS  
V04

PTESV_TYPO	= 00000016		
PTESV_TYP1	= 0000001A		
PTESV_VALID	= 0000001F		
QEMPTY	000000BC	R	05
RECONNECT	00000778	R R	05
RELDELPAGE	00000A71	R R	05
RELINIT	00000A57	R R	05
RELPAGE	00000A74	R R	05
RELPHD	00000316	R R	05
RPGCNT	00000010	R R	02
RSVAPTE	0000000C	R R	02
RWSSWP	00000008	R	02
SCH\$AQ_COMOH	*****	X	05
SCH\$CHSEP	*****	X X	05
SCH\$GB_SIP	*****	X X	05
SCH\$GL_COMOQS	*****	X X	05
SCH\$GL_CURPCB	*****	X X	05
SCH\$GL_FREECNT	*****	X X	05
SCH\$GL_FREELIM	*****	X X	05
SCH\$GL_MFYCNT	*****	X X	05
SCH\$GL_MFYLM	*****	X X	05
SCH\$GL_MFYLOLM	*****	X X	05
SCH\$GL_PCBVEC	*****	X X	05
SCH\$GL_SWPRATE	*****	X X	05
SCH\$GQ_COLPGWQ	*****	X X	05
SCH\$GQ_HIBWQ	*****	X X	05
SCH\$GW_DELPHDCT	*****	X X	05
SCH\$GW_QUAN	*****	X X	05
SCH\$GW_SWPFAIL	*****	X X	05
SCH\$GW_SWPFCNT	0000001A	R G	02
SCH\$NEQLVL	*****	X X	05
SCH\$OSWPSCHED	*****	X X	05
SCH\$V_MPW	*****	X X	05
SCH\$V_SIP	*****	X X	05
SCH\$W\$ITK	*****	X X	05
SETASTLVL	00000982	R R	05
SETUP	00000668	R R	05
SETWRTBAK	0000050E	R R	05
SGN\$GL_BALSETCT	*****	X X	05
SGN\$GL_FREEGOAL	*****	X X	05
SGN\$GL_FREELIM	*****	X X	05
SGN\$GL_PAGEDYN	*****	X X	04
SGN\$GL_PHDPAGCT	*****	X X	05
SPACEFAIL	00000254	R R	05
SUPER_MODE	000001D1	R R	04
SWAPEXIT	000009E1	R R	05
SWAPEXITA	000009E7	R R	05
SWAPRETRY	000009D7	R R	05
SWAPSCHEM	00000090	R	05
SWP\$GB_ISWPRI	*****	X X	05
SWP\$GB_PRI0	*****	X X	05
SWP\$GL_BALBASE	*****	X X	05
SWP\$GL_BALSPT	*****	X X	05
SWP\$GL_BSL0TSZ	*****	X X	05
SWP\$GL_HISWPCNT	*****	X X	05
SWP\$GL_HOSWPCNT	*****	X X	05
SWP\$GL_INPCB	*****	X X	05

SWP\$GL_ISPAGCNT	*****	X	05
SWP\$GL_ISWPCNT	*****	X X	05
SWP\$GL_ISWPPAGES	*****	X X	05
SWP\$GL_MAP	*****	X X	05
SWP\$GL_OSWPCNT	*****	X X	05
SWP\$GL_PHDBASVA	*****	X X	05
SWP\$GL_SHELIO	*****	X X	05
SWP\$GL_SHELLBAS	*****	X X	05
SWP\$GL_SWTIME	*****	X X	05
SWP\$GW_BALCNT	00000018	R G	02
SWP\$GW_IBALSETX	*****	X X	05
SWP\$SHELINIT	*****	X X	05
SWPREAD	00000A88	R	05
SWPWRITE	00000A8E	R	05
SYSS\$CRELMN	*****	X X	04
SYSS\$CREPRC	*****	X X	04
SYSS\$GB_DEFPRI	*****	X X	05
SYSTEM_ARG	000003D5	R R	04
SYSTEM_LIST	00000285	R R	04
SYSTEM_TABLE	000000C0	R R	03
SYSTEM_TABLE_LNMTH	000000E7	R R	03
SYSTEM_TABLE_ORB	00000110	R	03
SYSTEM_TABLE_ORB_SIZE	= 00000070		
SYSTEM_TABLE_SIZE	= 000000C0		
SYS_DISK_ARG	00000180	R	03
SYS_DISK_DESC	000000AD	R R	04
SYS-SYSDEVICE_ARG	00000198	R R	03
SYS-SYSDEVICE_DESC	000000BD	R R	04
TEMPORARY_MAILBOX_ARG	000003ED	R R	04
TEMPORARY_MAILBOX_LIST	000002A1	R R	04
TERMINAL_BUFFER	000001D9	R	04
TMP...	= 00000001		
TRNLOG_GS_ARG	00000405	R R	04
TRNLOG_GS_DESC	000000D2	R R	04
TRNLOG_GS_LIST	000002B1	R R	04
TRNLOG_PGS_ARG	0000044D	R R	04
TRNLOG_PGS_DESC	00000129	R R	04
TRNLOG_PGS_LIST	00000305	R R	04
TRNLOG_PG_ARG	0000041D	R R	04
TRNLOG_PG_DESC	000000EE	R R	04
TRNLOG_PG_LIST	000002CD	R R	04
TRNLOG_PS_ARG	00000435	R R	04
TRNLOG_PS_DESC	0000010B	R R	04
TRNLOG_PS_LIST	000002E9	R R	04
TTODESC	00000013	R	04
VAS\$VPN	= 00000015		
VAS\$V_SYSTEM	= 0000001F		
VAS\$V_VPN	= 00000009		
WSL\$C_SYSTEM	= 00000002		
WSL\$M_MODIFY	= 00000100		
WSL\$M_PAGTYP	= 0000000E		
WSL\$M_PFNLOCK	= 00000010		
WSL\$M_VALID	= 00000001		
WSL\$M_WSLOCK	= 00000020		
WSL\$V_MODIFY	= 00000008		
WSL\$V_PAGTYP	= 00000001		
WSL\$V_PFNLOCK	= 00000004		



SWAPPER  
Symbol table

WORKING SET SWAPPER

E 7

16-SEP-1984 01:19:37 VAX/VMS Macro V04-00  
5-SEP-1984 03:48:13 [SYS.SRC]SWAPPER.MAR;1

Page 58  
(28)

WSL\$V\_VALID  
WSL\$V\_WSLOCK  
WSLERR  
WSLOOP

= 00000000  
= 00000005  
00000526 R 05  
00000709 R 05

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABS\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$\$\$220	0000001C ( 28.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
\$\$\$260	000001B0 ( 432.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC 21
YF\$LOWUSE	00000638 ( 1595.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AEXENONPAGED	00000B26 ( 2854.)	05 ( 5.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
Z\$INIT\$PFN_FIXUP_TABLE	0000005A ( 90.)	06 ( 6.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:01.61
Command processing	127	00:00:00.50	00:00:04.66
Pass 1	515	00:00:22.43	00:01:06.36
Symbol table sort	0	00:00:02.75	00:00:05.98
Pass 2	423	00:00:06.86	00:00:20.84
Symbol table output	1	00:00:00.33	00:00:01.01
Psect synopsis output	0	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1103	00:00:32.98	00:01:40.51

The working set limit was 2250 pages.  
133641 bytes (262 pages) of virtual memory were used to buffer the intermediate code.  
There were 90 pages of symbol table space allocated to hold 1677 non-local and 138 local symbols.  
2492 source lines were read in Pass 1, producing 45 object records in Pass 2.  
38 pages of virtual memory were used to define 36 macros.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	20
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	13
TOTALS (all libraries)	33

1690 GETS were required to define 33 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SWAPPER/OBJ=OBJ\$:SWAPPER MSRC\$:SWAPPER/UPDATE=(ENH\$:SWAPPER)+EXECMLS/LIB



DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY